The analysis and evaluation of ISO/IEC9126–3 internal quality measures applicability: state-of-the-art 2006

Blanca Gil
Department of Software and IT Engineering
École de technologie supérieure,
1100 Notre Dame St. West, Montréal, Québec
H3C 1K3 Canada
Blanca-lucia.gil.1@ens.etsmtl.ca, blgc1@yahoo.com

Witold Suryn
Department of Software and IT Engineering
École de technologie supérieure,
1100 Notre Dame St. West, Montréal, Québec
H3C 1K3 Canada
witold.suryn@etsmtl.ca

Abstract: This white paper presents the results of the research project on the applicability and relevance of internal quality measures published in ISO/IEC 9126-3:2003 standard for software quality measurement and evaluation. As a result, the recommendations for modifications to the current set of ISO/IEC 9126-3 set of measures as well as to the existing quality model have been proposed.

Key words: Internal quality measures, quality model, Internal quality evaluation, Pure internal measures, measurement tracking.

INTRODUCTION

The objective of this white paper is to present the results of a revision of the ISO/IEC 9126 Part 3: Software Engineering - Product Quality - Internal Metrics and related documents, making emphasis on those subjects that remain in relation with the definition, identification, choice, documentation, evaluation, traceability, predictive capacity and practical uses of internal quality measures. In order to achieve this goal, a dedicated analysis methodology has been developed as the evaluation tool.

Also, due to the vastness of the subject the following limitations were applied when conducting the research:

- Only the ISO/IEC 9126–3 measures were studied using the officially published version of the standard,
- The analysis was applied from a software lifecycle and a stakeholder perspectives only,
- The verification of measures through practical measurements of artifacts was not a part of this research,
- The embedded software development technology, due to its specific nature, was not taken into account in this research.

The following results make the outcome of this research:

- The list of 42 research axiom definitions derived from ISO/IEC 9126-1, ISO/IEC 9126-3, and ISO/IEC 14598-1 [ISO01, ISO03,ISO04],
- The extended set of artifacts applicable as "input to measurements" for internal quality measures,
- A proposition of enhancements to the existing ISO/IEC 9126 quality model,
- The proposition of 32 new, 20 modified and 3 deleted internal quality measures,
- The enhanced prediction capacity tables reflecting the results of this research.

1. METHODOLOGY

To accomplish the objectives of this research a dedicated four-phase analysis methodology has been developed. The methodology is based on an in-depth analysis of the ISO/IEC 9126-3: Internal Quality Metrics standard [ISO03]. The below structure presents its components:

Phase I. Identification and definition of the axioms (derived from ISO/IEC 9126 part 1 and 3 and from ISO/IEC 14598 part 1),

Phase II. Construction of the set of basic hypotheses. The candidate hypotheses were identified through:

- Study of chosen software lifecycle process models,
- Study of chosen artifacts to be measured,
- Study of the current quality model and associated measures,
- Study of the current measurable properties of chosen types of software products,
- Study of ISO 9126-3 prediction capacity (from internal quality to external quality),

Phase III. Construction of the analysis engine. The following elements were taken into consideration:

- The measurement: when, what and for whom,
- The applicable axioms,
- The quality model structure,
- Inconsistencies in the prediction capacity of internal quality measures,

Phase IV. Analysis and recommendation for improvement of measures in categories of:

- Characteristic of a measure,
- Predictive capacity,
- Functional correctness, precision and adequacy of the definition,
- Understandability of the description.

2. ANALYSIS
2.1 Definition of axioms

2.1.1 Methodology

ISO/IEC 9126-1 - Quality model and ISO/IEC 14598-1 Software Product Evaluation - Part 1: General overview describe across their text fundamental features shared by all quality measures. In this paper those features - called further the *axioms* - are used as fundamental analysis criteria. To facilitate the reference to each one, the following axiom structure was used:

- The axiom's identification,
- The exact text taken from the Standard
- The location: the Standard and the page where the axiom is located inside the Standard (with P1 indicating ISO/IEC 9126-1, P3 - ISO/IEC 9126-3 and P5 - ISO/IEC 14598-1).

For the purpose of the analysis methodology all the axioms were grouped in three categories:

Category I: Scope, lifecycle and tracking (estimation and prediction),
Category II: Applicability and relevance,
Category III: Measurement – quality and nature of measurement.

2.1.2 The list of axioms

The thorough analysis of ISO/IEC 9126 part 1, part 3 and ISO/IEC 14598 part 1 has allowed for the identification of the following axioms:

Category I: Scope, lifecycle and tracking (estimation and prediction)

Axiom 1: The metrics listed in this International Technical Report are not intended to be an exhaustive set (P1.vi).
Axiom 2: The views of internal quality, external quality and quality in use change during the software lifecycle (P1.3).
Axiom 3: The fundamental nature of the software product quality represented by internal quality remains unchanged unless redesigned (P1.5).
Axiom 4: The current state of the art does not provide all the support necessary for the purposes of prediction (P1.5).
Axiom 5: [Internal metrics] can be used to predict values of the external metrics (P1.6).
Axiom 6: The correlation between internal attributes and external measures is never perfect and the effect […] will be determined by experience and will depend on the particular context […] (P1.14).
Axiom 7: It is often difficult to design a rigorous theoretical model that provides a strong relationship between internal metrics and external metrics (P3.3) – Axiom also in ISO/IEC 9126-1 (P1.15).
Axiom 8: Internal metrics should also have predictive validity, which means that they should correlate with some desired external measures (P1.17)
Axiom 9: The internal metrics standard shows the relationship between external and internal metrics (P5.8).
Axiom 10: Internal metrics are of most interest during the development process (P5.12).
Axiom 11: Internal metrics are of little value unless there is evidence that they are related to external quality (P5.13).

Category II: Applicability and relevance

Axiom 21: This report is applicable to any kind of software product, although each of the metrics is not always applicable to every kind of software product (P3.vi).
Axiom 22: Quality requirements cannot be completely defined before the beginning of design (P1.4).
Axiom 23: The hierarchy [characteristics and sub-characteristics] is not perfect, as some attributes may contribute to more than one sub-characteristic (P1.14).

Axiom 24: The internal metrics may be applied to a non-executable software product during its development stages (such as request for proposal, requirements, definition, design specification or source code) (3) – Axiom also in ISO/IEC 9126-1 (P3.15).
Axiom 25: […] internal metrics […] measure intrinsic properties including those, which can be derived form simulated behaviour (P1.15).
Axiom 26: The measurements of internal metrics use number or frequencies of software composition elements which appear for example on source code statements, the control graph, data flow and state transition representations (P1.15).
Axiom 27: Documentation can also be evaluated using internal metrics (P1.15).
Axiom 28: The metrics listed in this [metric table] are not intended to be an exhaustive set and may not have been validated (P3.4).
Axiom 29: Additional specific metrics for particular purposes are provided in other related documents, such as functional size measurements or precise time efficiency measurement (P3.4).
Axiom 30: Lines of code, complexity, the number of faults found in a walk through and the Fog Index are all internal measures made on the product itself (P5.4).
Axiom 31: Modularity and traceability are examples of internal attributes, which can be measured (P5.12).
Axiom 32: ISO/IEC 12207 SLCP Reference: identifies software lifecycle process(es) where the metric is applicable (P3.4).
Axiom 33: Target audience: Identifies the user(s) of the measurement results (P3.4).

Category III: Measurement

Axiom 40: Measurements should be objective, empirical using a valid scale, and reproducible (P1.16).
Axiom 41: Internal measure […] is not derived form measures of the behaviour of the system of which it is a part (P3.4).
Axiom 42: Lines of code, complexity, the number of faults found in a walk through and the Fog Index are all internal measures made on the product itself (P3.4).

2.2 Identification of basic hypotheses

Internal quality measures, types of software product recognized by the standard as "measurable", the required measurement input (artifacts) and prediction capacity of the measures are the core components of the applicability of the standard. The study of these components rendered the following results as basic research hypotheses:

Hypothesis 1: Reducing the applicability of measures to software lifecycle supporting processes severely minimizes the possibilities of developing a quality software product,
Hypothesis 2: The selection of artifacts to be measured could affect the applicability of measures,
Hypothesis 3: Current classification (relationship to quality model) of measures could affect their applicability,
Hypothesis 4: Properties of artifacts to be measured could affect the applicability of measures,
Hypothesis 5: Current inconsistencies in prediction capacity could affect the applicability of measures, and
Hypothesis 6: The actual set of internal quality measures may require modifications in order to improve the applicability of the standard.

2.2.1 Hypothesis 1: Reducing the applicability of measures to software lifecycle supporting processes severely minimizes the possibilities of developing a quality software product

According to axiom 32 (clause 3.1.2.2), ISO/IEC 9126-3 uses the ISO/IEC 12207 generic model as a referenced model of software lifecycle (Fig.1). The findings of this research have proven that the only references to software lifecycle processes present in internal quality measures tables are those made to supporting lifecycle processes from ISO/IEC 12207 (mostly Verification, Review and Validation). No primary (technical) processes are mentioned or referenced what suggests (as shown in Table 1) that the "recommended" actual applicability of internal quality measures does not cover the development phase (design, testing or coding) and its respective artifacts, thus limiting speed, range and efficiency of quality evaluation and its results (e.g. found deficiencies or recommended improvements of the evaluated software product).

Supporting lifecycle processes
6.1. Documentation process 6.2. Configuration management process 6.4. Verification process 6.5. Validation process 6.6. Joint review process
Development Processes
5.3.4. Software requirements analysis 5.3.5. Software architectural design 5.3.6. Software detailed design 5.3.7. Software coding and testing

Figure 1. ISO/IEC 12207 generic lifecycle process model

Table 1. ISO 9126-3: ISO/IEC 12207 Process references

ISO/IEC 12207 process	Number of ISO 9126-3 measures referenced	Percentage
Verification	60	86 %
6.6 Joint Review	60	86 %
6.5 Validation	13	19 %
6.8 Problem Resolution	4	6 %
6.3 Quality Assurance	2	3 %
5.3 Quality Testing	1	1 %
5.4 Operation	1	1%

Such a situation, if retained, remains in an open conflict with the applicability objectives of the standard stated in ISO/IEC 9126-1 Annex A.1.2 page 15:

Internal metrics can be applied to a non-executable software product (such as a specification or source code) during designing and coding. When developing a software product the intermediate products should be evaluated using internal metrics which measure intrinsic properties, including those which can be derived from simulated behaviour.

At the same time ISO/IEC 12207 explicitly indicates the requirement of creating appropriate documentation within each of the software lifecycle processes, both primary and supporting, making it a perfect candidate for "input to measurement".

From the perspective of the user of the standard as well as from the perspective of better applicability of internal quality measures it is then reasonable to combine all appropriate ISO/IEC 12207 processes and their respective artifacts with corresponding internal quality measures. We need to build a clear link between the measure, the input to measurement and the point of its collection, i.e. the process and its phase with focus rather on primary (technical) processes than the supporting ones. Such a link will also help better identify "targeted audience", (Table 2 shows this column content), or participants, as different processes require different type of expertise.

Table 2. ISO 9126-3 Target audiences

ISO 9126-3 Target audiences	Number of measures referenced	Percentage
Developers	68	97 %
Requirers	54	77 %
Maintainers	22	31 %
Users	2	3 %

2.2.2 Hypothesis 2: Selection of artifacts to be measured could affect the applicability of measures

According to the data found in ISO 9126-3 column "Inputs to measurement" (Table 3 shows the percentage of artifacts use this columns), Review Report, Requirements Specification, Design and Source Code are the most

important set of data inputs for measurement process. However, even if existing, this information is in most cases imprecise. The following lacking elements were found:

- Unclear definition of content and scope of Review Report. In software engineering domain any artefact created during a development process can be the subject of a review. Moreover, the artifacts created in different phases of the process seriously differ, what makes the recommendation of using Review Report meaningless, unless a precise indication of what product should be reviewed is given. This observation remains in close correlation with findings illustrating the hypothesis 1.

- The usability ISO 9126-3 Pure Internal Metrics strongly depends on the precise indication of required artifacts, being in most cases of the technical nature (for example, "Conditional statement" or "Program size" require a specific software artefact – source code). A quick look at the existing Pure Internal Metrics table reveals that not only indication of artifacts is missing but also a reference to software lifecycle processes or even targeted audience have not been considered worth of recommending.

Table 3. ISO 9126-3 Inputs to measurement

ISO 9126-3 Inputs to measurement	Number of measures referenced	Percentage
Review Report	60	86 %
Requirements Specifications	49	70 %
Design	48	69 %
Source Code	24	34 %
Standards & Regulations	6	9 %
Know operation system	4	6 %
Estimated time in system calls	4	6 %
Organization Data Base	1	1 %
Fault Removal Report	1	1 %
Test Plan	1	1 %
Estimated size memory utilization	1	1 %
Configuration control system	1	1 %
Version logs specifications	1	1 %
Test report	1	1 %

As the above hypothesis has proven valid the extended set of artifacts addressing this problem has been identified. The new list of possible artifacts is presented below together with their applicable conditions:

a. If nothing else is indicated, the generic input to measurement could be chosen from:
   - Joint Review Report,
   - Validated Requirement Specification Documents,
   - Standards and Regulations,
   - Verification report.

b. If nothing else is indicated or none specific product is pointed, the products to be measured could be chosen from:
   - Feature Model,
   - Functional Model,
   - Data Model,
   - Event Model,
   - Hardware Model,
   - Source code,
   - User documentation,
   - Technical documentation,
   - Technical documentation,
   - Use Case Diagram,
   - Activity Diagram,
   - State Diagram,
   - Sequence Diagram,
   - Class/object Diagram,
   - Component Diagram,
   - Deployment Diagram

The above list does not claim to be complete or exhaustive; however its content tries to reflect the actual reality of software engineering where object-oriented development seems to gain the dominant position.

2.2.3 Hypothesis 3 and 4: Current measures classification and properties of artifacts to be measured could affect the applicability of measures

During the analysis of the standard several inconsistencies in the classification of measures were found. The important example is the ISO/IEC 9126-3 table of "Pure internal metrics". These metrics are very technical, with obvious relationship to software quality, in particular to higher level Internal quality measures, but are mentioned as "informative" in Annex E with no appearance in ISO/IEC 9126 quality model. Due to their unquestionable importance it seemed profitable to incorporate them into the model through a modification of its existing structure, thus indicating their presence and applicability.
The enhanced model with added new characteristic named "Internal technical measures" is presented in Fig. 2.

The proposed "Internal technical measures" characteristic consists of:

- "Development standard measures" sub-characteristic assessing the adherence to writing guidelines for product components. This sub-characteristic would contain:
    - Self-descriptiveness measures: measures of the product attributes that describe the product by itself,
    - Self-containedness measures: measures of the product elements that allow understanding of the product nature by itself,
    - Pure reusability measures: measures of the extent to which a product can be used in contexts different from these it was originally designed for,
    - Pure maintainability measures: measures of the extent to which the software can be modified at the lowest possible cost,
- "Size" sub-characteristic providing measures of a size of a product (for example lines of code or functional points),
- "Complexity" sub-characteristic with measures assessing complexity of a product.

This extended quality model offers some important benefits to the users of the standard however the potential application of proposed changes requires careful consideration of possible impact on the existing use of the standard. The possible solutions are discussed in clause 3. However, in Tables 4 and 5, we present a suggested set of Internal technical measures that will be referenced by the modified ISO/IEC 9126 Internal quality measures.

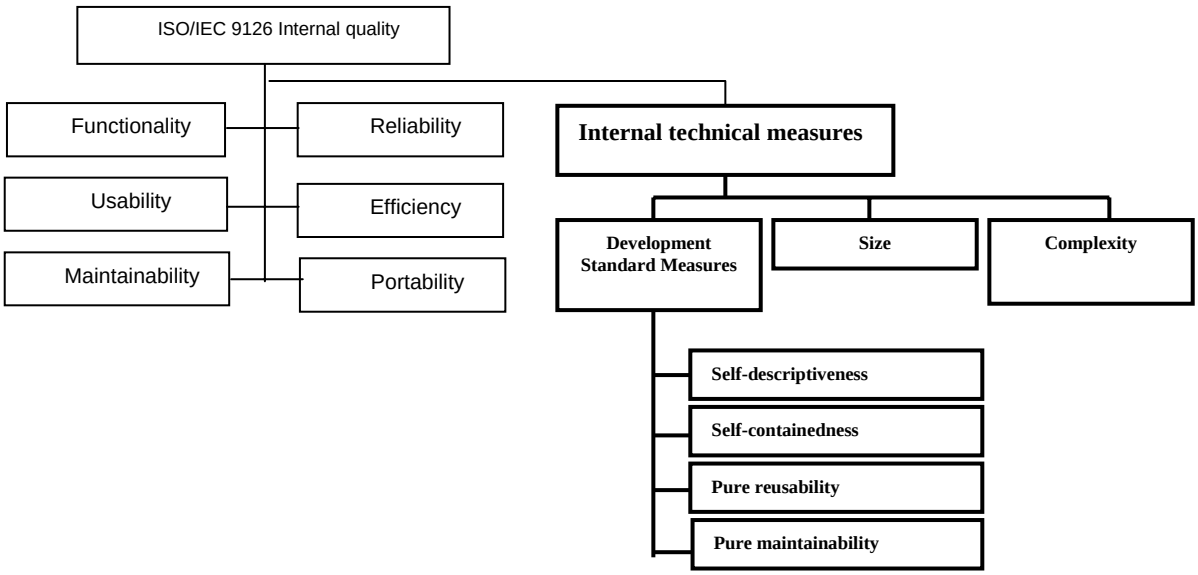

Figure 2. Enhanced ISO/IEC 9126 Internal quality model

Table 4. Suggested internal technical quality measures: general applicability.

Measure Name	Purpose of the measure	Measurement, formula and data element computations	Interpretation	Measure scale type & measure type
Coherence				
Coherence Traceability	To measure effectiveness of documentation and design structure and code of software product in mapping functions from requirements to implementation.	X=A/B A=Number of traceable items confirmed in review B=Number of items checked	0 <= X <= 1 The closer to 1, the better.	Absolute X=count/count A=count B=count
Complexity				
Cyclomatic number	To measure the level of complexity of the software design and coding structure	e-n+2p e: # of sides n: # of edges p: # of adjacent components		
Information Flow Complexity	To measure complexity of design control structure. (refer to IEEE 982.1)	IF C(Information Flow Complexity) = (fanin x fanout)2		
Self-descriptiveness				
Modularity	To measure the easiness to update and generalize the functional knowledge base on program function/data, sequence of execution, and hierarchy of control flow.	X1=A1/B1 X2=A2/B2 A1=the number of modules that are functionally associated with each other B1=the number of modules A2=the number of modules that are associated with each other in data structure B2=the number of modules		
Function-coupled module ratio	To measure the function-coupled module ratio	X=A/B A=Number of function-coupled modules confirmed in review B=Total number of all modules	0 <= X <= 1 The closer to 1, the better.	Absolute X=count/count A=count B=count
Size				
LOC	To measure the program physical size	X= number of Lines of Code	0 <= X	X=size
Program size	To measure the program scale	(N1+N2)log2(n1+n2) N1: operator occurrences N2: operand occurrences n1: total # of operators n2: total # of operands		
Program statements	To measure the program source statement	X=A A=Total number of program statements	0 <= X	X=size A=size
Average module size	To measure the average module size	X=A/B A=Total lines of source statements in all modules B=Total number of all modules	0 <= X	Absolute X=size A=size
Self-containedness				
Conditional statement	To measure the complexity level of coded modules	X=A A= Number of conditional statements	0 <= X	X=size A=size
Unified data reference	To measure the data unification	X=A/B A=Number of data references with unified name confirmed in review B=Total number of data references	0 <= X <= 1 The closer to 1, the better.	Absolute X=count/count A,B=count
Adequacy of variable names	To measure the variable names adequacy	X=A/B A=Number of variables with adequate names confirmed in review B=Total number of variables	0 <= X <= 1 The closer to 1, the better.	Absolute X=count/count A,B=count
Data-coupled module ratio	To measure the data-coupled module ratio	X=A/B A=Number of data-coupled modules confirmed in review B=Total number of all modules	0 <= X <= 1 The closer to 1, the better.	Absolute X=count/count A,B=count

Table 5. Suggested internal technical quality measures: object-oriented applicability..

Measure Name	Measurement, formula and data element computations	Interpretation
OO Complexity		
Average Method Complexity	X=Sum of all methods' cyclomatic complexity / Total number of application methods	X>=0, bigger X, more difficult to maintain, to comprehend to test and less reliability.
Degree of Cohesion of Objects	Total Fan-in for All Objects / Total No. of Objects	X>=0, bigger X, low degree of errors, less complex and more reliability.
OO Size		
Methods per Class	X=Total number of methods / Total number of object classes	X>=0, bigger X, less testability, more reuse.
Weighted Methods per Class	X= sum of the complexities of all methods of a class	X>=0, bigger X, less maintainability, less reusability.
OO Maintainability and reusability		
Application Granularity	X=Total number of objects / total function points	X>=0, bigger X, lower number of functions per object, more easily maintained, more reusable, more easily understood and analyzed
Degree of Coupling Between Objects	Average number of uses dependencies per object = total number of arcs / total number of objects Arcs = max (number of uses arcs) - in an object uses network arcs - attached to any single object in a uses network	X>=0, bigger X, lower maintainability, reuse, testability and error generation.
Factoring Effectiveness	X=Number of Unique Methods / Total Number of Methods	X>=0, bigger X, more reliability, maintainability, less coding errors, reusability
Coupling between object classes	X= the count of the classes to which this class is coupled. Two classes are coupled when methods declared in one-class use methods or instance variables of the other class.	X>=0, bigger X, less reusability, less modifiability, less testability.
Response For a Class	X= number of methods in the set of all methods that can be invoked in response to a message sent to an object of a class	X>=0, bigger X, less testability, more complexity
Lack of Cohesion in Methods	X= the number of different methods within a class that reference a given instance variable.	X>=0, bigger X, more complexity
OO Reusability		
Degree of Reuse of Inheritance Methods	Percent of Potential Method Uses Actually Reused (PP): PP = (Total Number of Actual Method Uses / Total Number of Potential Method Uses) x 100 X=PP Percent of Potential Method Uses Overridden (PM): PM = (Total Number of Methods overridden / Total Number of Potential Method Uses) x 100 X=PM	X>=0, bigger X, more reusability
Object Library Effectiveness	X= Total Number of Object Reuses / Total Number of Library Objects	
Inheritance Dependencies	X= max (inheritance tree path length)	X>=0, bigger X, more reusability, less testability, less compressibility.
Depth of Inheritance Tree	X= the maximum length from the node to the root of the tree	X>=0, bigger X, more complex, reusability
Number of Children	X= the number of immediate subclasses	X>=0, bigger X, more reuse, less testability

2.2.4 Hypothesis 5: Current inconsistencies in prediction capacity could affect the applicability of measures

According to ISO 9126 axioms 8, 9 and 11 ("Internal metrics are of little value unless there is evidence that they are related to external quality", clause 3.2.1.1), the prediction capacity of internal measurements is a fundamental applicability feature. In course of this research the prediction relationship between internal and external quality measures has been analyzed using the following qualification rules:

a) Sharing the same name and purpose. If measures shared both name and purpose the analysis concluded that a direct prediction relationship exists,
b) Sharing the same measurement purpose only. If measures shared only the purpose it was concluded that the internal measure could predict the results of one or more external measurements and that there is an indirect prediction relationship,
c) No sharing. If measures shared neither the name nor the purpose in any sense, it was concluded that there was no visible prediction relationship.

The statistical data of internal/external quality measures predictability relationship are presented in Table 6.

When analyzing the statistics of prediction capacity of internal quality measures the following observations were drawn:

- The nature of prediction is non-orthogonal, i.e. there are internal quality measures that are related to more than one external quality measure. For example, in Reliability characteristic the measure of fault detection is found being indirectly related to five external quality measures,
- The number of internal quality measures that remain non-related (or, in other words, do not serve the predictability purpose) is relatively small (7 out of 70, or 10%),
- The number of external quality measures that have no visible relation to internal quality measures is much greater (28 out of 112, or 25%)

Table 6. Statistics of prediction relationship between Internal and External quality measures

Characteristic (Internal and External Quality)	Number of all Internal quality measures	Number of all External quality measures	Internal quality measures with direct prediction relationship	Internal quality measures with indirect prediction relationship	Internal quality measures with no prediction relationship	External quality measures with no prediction relationship
Functionality	14	14	12	3	2	0
Reliability	8	18	6	7	0	1
Usability	18	28	7	10	0	7
Efficiency	9	24	5	3	2	15
Maintainability	9	16	2	6	1	4
Portability	12	12	10	1	2	1

All the three above observations indicate that the prediction mechanism between internal and external quality is not complete, leaving too many external and internal quality measures/attributes unrelated to each other. The details of the summary from Table 6 are presented in following Tables 7 to 12, for which the below legend applies:

- the different colors mean:

	It exits a direct relationship between internal and external measures.
	No exact relationship between internal and external measures but it is proposed an indirect relationship.
	No exact relationship between internal and external measures and it is not possible to propose an indirect relationship with the current ISO 9126 measurements.

- the abbreviations mean

X marks the internal and external measures intersection.
A marks the suggested internal measures.
C marks Internal quality measures with indirect prediction relationships.

Table 7. ISO 9126 Current Functionality tracking

FUNCTIONALITY	**External Measurements**													
Internal Measurements														
	Functional adequacy	Functional implementation completeness	Functional implementation coverage	Functional specification stability	Accuracy to expectation	Computational accuracy	Precision	Data exchangeability (Data format based)	Data exchangeability (User's success based)	Access auditability	Access controllability	Data corruption prevention	Functional Compliance	**Interface** standard compliance
Functional adequacy	**X**				**A**									
Functional implementation completeness		**X**			**A**									
Functional implementation coverage			**X**		**A**									
Functional specification stability				**X**										
Computational accuracy						**X**								
Precision							**X**							
Data exchangeability								**X**	**X**					
Interface consistency (protocol)														
Access auditability										**X**				
Access controllability											**X**			
Data corruption prevention												**X**		
Data encryption														
Functional Compliance													**X**	
Intersystem standard compliance														**X**

Table 8. ISO 9126 Current Reliability tracking

RELIABILITY	**External Measurements**																
Internal Measurements																	
	Estimated latent fault density	Failure density against test cases	Fault density	Failure resolution	Fault removal	Mean time between failures	Test coverage	Test maturity	Breakdown avoidance	Failure avoidance	Incorrect operation avoidance	Availability	Mean down time	Mean recovery time	Restartability	Restorability	Restore effectiveness	Reliability compliance
Fault detection	**C**	**C**	**C**					**C**				**C**						
Fault removal				**A**	**X**			**A**				**A**						
Test adequacy							**C**	**C**				**C**						
Failure avoidance									**A**	**X**								
Incorrect operation avoidance									**A**		**X**							
Restorability													**A**	**A**	**A**	**X**		
Restoration Effectiveness													**A**	**A**	**A**		**X**	
Reliability Compliance																		**X**

Table 9. ISO 9126 Current Efficiency tracking

EFFICIENCY	**External Measurements**																									
I. Measurements																										
	Response time	Mean time to response	Worst case response time ratio	Throughput time	Mean amount of throughput	Worse case throughput ratio	Turnaround time	Mean time for turnaround	Worst case turnaround time ratio	Waiting time	I/O device Utilization	Mean I/O fulfillment radio	User waiting time of I/O devices utilization	I/O related errors	I/O loading limits	Mean occurrence of memory error	Ratio of memory error / time	Maximum memory utilization	Mean occurrence of transmission error	Transmission capacity utilization	Mean of transmission error / time	Maximum transmission utilization	Media device utilization balancing	Efficiency Compliance
Throughput time				**X**	**A**																			
Response time	**X**																							
Turnaround time							**X**																	
I/O Utilization											**X**													
I/O Utilization Message density																								
Memory utilization																		**C**						
Memory utilization message density																								
Transmission Utilization																				**C**		**C**		
Efficiency Compliance																								**X**

Table 10. ISO 9126 Current Usability tracking

USABILITY	**External Measurements**																										
I. Measurements																											
	Completeness of description	Demonstration accessibility	Demonstration accessibility in use	Demonstration effectiveness	Evident functions	Function understandability	Understandable input and output	Ease of function learning	Ease of learning to perform a task in use	Effectiveness of user documentation and help	Effectiveness of user documentation in use	Help accessibility	Help frequency	Operational consistency in use	Error correction	Error correction in use	Default value availability in use	Message understandability in use	Self explanatory error messages	Operational error recoverability in use	Time between human error operation in use	Undoability	Customizability	Operation procedure reduction	Physical accessibility	Attractive interaction	**User** Interface appearance customizability	Usability Compliance
Completeness of description	**X**																										
Demonstration capability		**C**	**C**	**C**																							
Evident functions					**X**																						
Function understandability						**X**																					
Completeness of user documentation and/or help facility										**C**	**C**	**C**															
Input validity checking															**C**	**C**											
User operation cancelability															**C**	**C**											

User operation Undoability														**C**	**C**						**C**							
Customizability																						**X**					
Physical accessibility																							**X**				
Operation status monitoring capability														**C**	**C**												
Operational consistency													**C**	**C**	**C**												
Message Clarity																	**C**	**C**									
Interface element clarity														**C**	**C**												
Operational error recoverability														**C**	**C**				**C**								
Attractive interaction																									**X**		
User Interface appearance customizability																										**X**	
Usability Compliance																											**X**

Table 11. ISO 9126 Current Maintainability tracking

MAINTAINABILITY	**External Measurements**															
Internal Measurements	0															
	Diagnostic function support	Audit trail capability	Failure analysis efficiency	Failure analysis capability	Status monitoring capability	Software change control capability	Parameterized modifiability	Modification complexity	Change cycle efficiency	Change implementation elapse time	Change success ratio	Modification impact localization	Re-test efficiency	Availability of built-in test function	Test restartability	Maintainability Compliance
Activity recording	**C**	**C**	**C**	**C**	**C**											
Readiness of diagnostic function	**C**			**C**	**C**											
Change recordability						**C**										
Change impact											**C**					
Modification impact localization												**X**				
Completeness of built-in test function													**C**	**C**		
Autonomy of testability																
Test progress observability													**C**		**C**	
Maintainability Compliance																**X**

Table 12. ISO 9126 Current Portability tracking

PORTABILITY	**External Measurements**											
Internal Measurements												
	Adaptability of data structures	Organizational Environment adaptability	Hardware Environmental Adaptability	System software Environmental adaptability	Porting User Friendliness	Ease of installation	Ease of **set-up re-try**	Available co-existence	Continued use of Data	Functional inclusiveness	User support functional consistency	Portability Compliance
Adaptability of data structures	**X**											
Organizational Environment adaptability		**X**										
Hardware Environmental Adaptability			**X**									
System software Environmental adaptability				**X**								
Porting User Friendliness					**X**							
Ease of **setup retry**						**A**	**X**					
Installation effort												
Installation flexibility												
Continued use of Data									**X**			
Functional inclusiveness										**X**		
Available co-existence								**X**				
Portability Compliance												**X**

The analysis of the relationship between internal and external quality measures shown in tables 7 to 12 revealed possibilities of addressing the problem of incompleteness of prediction mechanism between the two categories of quality, however mostly through the addition of new measures discussed in section 2.2.5 below.

2.2.5 Hypothesis 6: The actual set of internal quality measures may require modifications in order to improve the applicability of the standard

As the ISO/IEC 9126 standard has undergone continuous development since 1991 it is understandable that the actual measures, especially in internal quality area, where the technological evolution is the fastest may prove partially incomplete, obsolete or even unnecessary. This part of the research had as the objective to verify how the existing set (and nature) of internal quality measures serve their users in their direct application during development of a software product.

The following criteria were applied when analyzing the existing measures:

- Characteristic of an attribute. This criterion verifies whether the measure of a given attribute fits into its actual sub-characteristic in the quality model.
- Predictive capacity. Predictive capacity criterion verifies an existence and the type of prediction relationship of the analyzed internal quality measure to its possible counterparts in external quality.
- Definition and description of the measure. This criterion verifies functional correctness, precision, adequacy and understandability of definitions and descriptions of measures in order to find eventual applicability problems resulting from wrong usage of measures.

The obtained results presented in Tables 13, 14 and 15 seem to be self-explanatory, but the below comments may prove helpful:

- Most of additions were proposed to the Maintainability and the Usability sub-characteristics (12 and 10 respectively), what may suggest that different aspects of usability might require some evolutionary

corrections. One of possible reasons for such a state could be a de-synchronization between speed of developing the standard and dynamics of technological evolution in the domain,

- Most of modifications were recommended for Functionality and the Efficiency sub-characteristics (number of affected measures: 10 and 5 respectively; number of singular modifications: 31 and 20 respectively). The nature of possible modifications indicates rather missing precision than serious lacks of information,
- Only 3 measures were proposed to be deleted (Efficiency - 2 and Reliability - 1) which may be interpreted as a positive indicator of the generic validity of the core of quality model and related internal quality measures.

Table 13. Internal quality measures – identified candidates for inclusion

FUNCTIONALITY	
1	Computational completeness
2	Precision accuracy
3	Data exchangeability completeness (data format based)
USABILITY	
5	Demonstration consistency
6	User Interface consistency
7	User Documentation consistency
8	Output Messages consistency
9	Help consistency
10	Help understandability
11	Completeness of training material
12	Default value availability
13	Operation procedure reduction
14	Consistency with others known systems
EFFICIENCY	
27	Estimated Software platform response time
28	Estimated Data management response time
29	Estimated Transmission response time
30	Estimated I/O Utilization size

RELIABILITY	
4	Fault density
MAINTAINABILITY	
15	Activity recording legibility
16	Modification complexity
17	Parameterized availability
18	Reuse utilization
19	Programming Style consistency
20	Frameworks utilization
21	Patterns utilization
22	Programs libraries utilization
23	Data stores / procedures utilization
24	Technical Documentation consistency
25	Technical Documentation understandability
26	Completeness of Technical Documentation
PORTABILITY	
31	Internationalization
32	Ease of installation

Table 14. Internal quality measures – identified candidates for deletion

RELIABILITY	
1	Incorrect operation avoidance

EFFICIENCY	
2	Turnaround time
3	Transmission Utilization

Table 15. Internal quality measures with identified needs for modifications

FUNCTIONALITY		**EFFICIENCY**	
1	Functional adequacy	12	Response time
2	Functional implementation completeness	13	Throughput time
3	Functional implementation coverage	14	I/O utilization
4	Computational accuracy	15	I/O utilization message density
5	Access auditability	16	Memory utilization message density
6	Data corruption prevention	**RELIABILITY**	
7	Intersystem standard compliance	17	Fault detection
8	Precision	18	Test adequacy
9	Data exchangeability	19	Failure avoidance
10	Interface consistency	20	Restoration effectiveness
USABILITY			
11	Demonstration capability		

The aggregated results obtained by this analysis are presented in the below tables (Table 16 and 17), and details are given in Table 18.

Table 16. Statistics of interventions in individual measures area

Characteristics	Additions	Major Modifications	Minor Modifications	Deletions
Functionality	3	20	11	0
Reliability	1	5	3	1
Usability	10	1	0	0
Efficiency	4	16	4	2
Maintainability	12	0	0	0
Portability	2	0	0	0
T O T A L = 95	**32**	**42**	**18**	**3**

Table 17. Classification of interventions in individual measures area

Characteristic	CRITERIA											
	Attribute Characteristic				Description and Definition				Predictive Capacity			
	Mi	My	A	D	Mi	My	A	D	Mi	My	A	D
Functionality			3		11	20			1	6	2	
Reliability	1	1	1	1	1	2					1	
Usability			10			1				1	10	
Efficiency	2	14	4	2	3				2	7	3	1
Maintainability			11								4	
Portability			2								1	
T O T A L	**3**	**15**	**29**	**3**	**15**	**23**	**0**	**0**	**3**	**14**	**21**	**1**

Table 18. Classification of interventions in individual measures area (detailled)

Sub-characteristic	CRITERIA															
	Attribute Character				Semantic				Predictive Capacity				Product Type			
	Mi	My	A	D	Mi	My	A	D	Mi	My	A	D	Mi	My	A	D
Functionality			**3**		**11**	**20**			**1**	**6**	**2**					
Suitability					6	2										
Accuracy			2		2	5					2					
Interoperability			1		1	6				1						
Security					2	6			1	4						
Compliance						1				1						
Reliability	**1**	**1**	**1**	**1**	**1**	**2**					**1**		**1**	**2**	**1**	
Maturity	1	1	1		1	2					1				1	
Fault tolerance				1									1	1		
Recoverability														1		
Compliance																
Usability			**10**			**1**				**1**	**10**				**1**	
Understandability			4			1				1	4					
Learnability			3								3				1	
Operability			2								2					
Attractiveness																
Compliance			1								1					
Efficiency	**2**	**14**	**4**	**2**	**3**				**2**	**7**	**3**	**1**	**2**	**16**		
Time behavior	2	7	3	1	2				2	7	3		2	9		
Resoruce utilization		7	1	1	1							1		7		
Compliance																
0Maintainability			**11**								**4**				**1**	
Analyzability			1								1					
Changeability			8								3				1	
Stability			3													
Testability																
Compliance																
1Portability			**2**								**1**					
Adaptability			1													
Installability			1								1					

Replaceability																
Co-existence																
Compliance																
T O T A L	**3**	**15**	**29**	**3**	**15**	**23**	**0**	**0**	**3**	**14**	**21**	**1**	**3**	**18**	**1**	**0**

Where:

- Mi – minor modification
- My – important modification
- A – addition
- D – deletion

3. FULL ANALYSIS

The following tables show 56 analysis modules obtained in course of the presented research program. Only ISO 9126-3 measures that, according to the research results, require improvement suggestions are discussed in the tables. The below legend for the column "Type & Action" applies:

- Type:
  - **AC**:According to the attributes characteristics.
  - **PC**: According to the predictive capacity (from internal quality to external quality),
  - **SM**: According to the semantic of the measurement.
  - **PT**: According to the product type.

- Action (explained in the column "Suggestion"):
  - **Add**: Addition. When a new measure could be inserted.
  - **Mod**: Modification. When a definition of a current measure could be changed.
  - **Del**: Elimination. When a current measure could be rejected.

3.1 Functionality measures

3.1.1 Suitability measures

3.1.1.1 (1) Functional adequacy

Problem	Type & Action	Suggestion
Meaning of "adequacy" in "Method of application"	SM Mod	Add the comment: "Synonymous of adequacy: satisfactory or acceptable".
Meaning of "function" in "Method of application"	SM Mod	Add the comment: "Function or functional item may be a Functional Requirement, a Module, a Procedure, a programmable Function, a Stored Procedure, a Use Case, a Class, a Class Method, an Object or an Object Method, descriptions and source code (it is not an exhaustive list)".

3.1.1.2 (2) Functional implementation completeness

Problem	Type & Action	Suggestion
Meaning of "completeness" in "Method of application"	SM Mod	Add the comment: "Synonymous of completeness are: having all the necessary parts or entire".
Meaning of "implementation" in "Method of application"	SM Mod	Add the comment: "Implementation or a function item must be implemented with a Function Model, Data Model or Source Code".

3.1.1.3 (3) Functional implementation coverage

Problem	Type & Action	Suggestion
It is Accuracy not a Functionality measurement.	AC Mod	Translate this measurement to the sub-characteristic "Functionality / Accuracy". **This suggestion is not essential. No change was made.**
Weak coherence between "Metric Name" and "Purpose of the metrics"	SM Mod	Change "Metric Name" of "Functional implementation coverage" to "Functional implementation correctness"
Meaning of "correctness" in "Method of application"	SM Mod	Add the comment: "Synonymous of correctness are: free from error or true or right".
Meaning of "function item" in "Method of application" column.	SM Mod	Eliminate the note "Review by function item" in "Method of application" column because it was defined in a global comment (see "Functional adequacy" measurement).
Duplication of measurement in "Method of application" column.	SM Mod	Eliminate the text "or missing functions" because "3.Functional implementation completeness" measure already uses that method.

3.1.1.4 (4) Functional specification stability (volatility)

Problem	Type & Action	Suggestion
It is Maturity measurement not a Functionality measurement.	AC Del	Translate this measurement to the sub-characteristic "Reliability / Maturity". **This suggestion is not essential. No change was made.**

3.1.2 Accuracy measures.

3.1.2.1 (5) Computational accuracy

Problem	Type & Action	Suggestion
Meaning of "computational accuracy" in "Method of application"	SM Mod	Add the comment: "Computational synonymous: mathematical calculation, accuracy synonymous: correct in all details".
Weak coherence between "Metric Name" and "Purpose of the metrics"	SM Mod	Change the "Purpose of the metrics" to "How correct have the mathematical calculation been implemented".
Weak coherence between "Metric Name" and "Method of application"	SM Mod	Change the "Method of application" to "Count the number of mathematical calculations that have been correctly implemented and compare with the number of mathematical calculations described in requirement specifications".
Weak coherence between "Metric Name" and "Measurement, formula and data element computations"	SM Mod	Change the "Measurement, […]" to "X=A/B A=Number of accurate mathematical calculations as confirmed in evaluation. B=Number of mathematical calculations described in requirement specifications".
Weak coherence between "Metric Name" and "Interpretation of value"	SM Mod	Change the "Interpretation of measured value" to "0<=X<=1. The closer to 1, the more accurate".

3.1.2.2 (6) Computational completeness

Problem	Type & Action	Suggestion
No measure for Computational completeness or a measure to predict the external measure: "Computational accuracy" as the "Purpose" column explains.	AC, PC Add	Add the measure "Computational completeness".
• Purpose of the measure:		How complete have the mathematical calculations been implemented?
• Method of application:		Count the number of mathematical calculations that have been implemented and compare with the number of mathematical calculations described in requirement specifications.

3.1.2.3 (7) Precision

Problem	Type & Action	Suggestion
Weak coherence between "Metric Name" and "Purpose of the metrics"	SM Mod	Change the "Metric Name" to "Precision completeness".
Meaning of "precision" in "Method of application"	SM Mod	Add the comment: "Data item (data): an elementary piece of information. Precision: valid range of data type units. Data type examples: integer, real, date, string, etc."

3.1.2.4 (8) Precision accuracy

Problem	Type & Action	Suggestion
No measure for Precision accuracy or a measure to predict external measure: "Precision" as the "Purpose" column explains.	AC, PC Add	Add the measure "Precision accuracy".
• Purpose of the measure:		How correct have the precision of data items been implemented?
• Method of application:		Count the number of data items that have been implemented with the incorrect level of precision and compare with the total number of data items with specific level of precision described in requirement specifications.

3.1.3 Interoperability measures.

3.1.3.1 (9) Data exchangeability (data format based)

Problem	Type & Action	Suggestion
Weak coherence between "Metric Name", external measure "Data exchangeability (data format based)" and "Purpose of the metrics"	SM, PC Mod	Change the "Purpose of the metrics" to "How correct have the data formats used as interface with other software or system been implemented?"
Weak coherence between "Metric Name" and "Purpose of the metrics"	SM Mod	Change the "Metric Name" to "Data exchangeability accuracy (data format based)".

3.1.3.2 (10) Data exchangeability completeness (data format based)

Problem	Type & Action	Suggestion
No measure for Data exchangeability completeness	AC Add	Add the measure "Data exchangeability completeness (data format based)".
• Purpose of the measure:		How complete have the data formats used as interface with other software or system been implemented?
• Method of application:		Count the number of data formats used as interface that have been implemented and compare with the number of data formats used as interface described in requirement specifications.

3.1.3.3 (11) Interface consistency (protocol)

Problem	Type & Action	Suggestion
Meaning of "consistency" in "Method of application"	SM Mod	Add the comment: "Consistency synonymous: conforming to a regular pattern or unchanging".
Weak coherence between "Metric Name" and "Purpose of the metrics"	SM Mod	Change the "Purpose of the metrics" to "How correct, complete and unchanging have the interface protocols been implemented?"
Weak coherence between "Metric Name" and "Purpose of the metrics"	SM Mod	Change the "Metric Name" to "Interface protocol consistency"
Weak coherence between "Metric Name" and "Method of application"	SM Mod	Change the "Method of application" to "[…] correctly, complete and unchanged […]"
Weak coherence between "Method of application" and "Measurement, formula"	SM Mod	Change the "Measurement, formula" to "[…] A=Number of interface protocol that have been implemented correctly, complete and unchanged as detected in evaluation […]"

3.1.4 Security measures.

3.1.4.1 (12) Access auditability

Problem	Type & Action	Suggestion
Weak coherence between "Metric Name", external measure "Access auditability" and "Purpose of the metrics"	SM, PC Mod	Change the "Purpose of the metrics" to "How auditable is access to the system?"
Weak coherence between "Purpose of the metrics" and "Method of application"	SM Mod	Change the "Method of application" to changing "logged" for "recorded".
Weak coherence between "Purpose of the metrics" and "Measurement, formula"	SM Mod	Change the "Measurement, formula" to changing "logged" for "recorded".
Clarify the elimination of "logged" action.	SM Mod	Add the comment: "Login is one of the access types to be recorded".

3.1.4.2 (13) Data corruption prevention

Problem	Type & Action	Suggestion
Weak coherence between "Method of application" and external measure "Data corruption prevention"	SM, PC Mod	Change the "Method of application" to "[…] and compare with the number of illegal operations specified in requirements [...]"
Weak coherence between "Measurement, formula" and external measure "Data corruption prevention"	SM, PC Mod	Change the "Measurement, formula" to "[…] B=Number of illegal operations specified in requirements."
Meaning of "illegal operation" in Weak coherence between "Method of application"	SM, PC Mod	Add the comment: "Illegal operations: operations capable of corrupting or destroying data."
Weak coherence between "Method of application" and "Interpretation of value"	SM, PC Mod	Change the "Interpretation of value" to "[…] The closer to 1, the more preventive."

3.1.5 Compliance for Functionality measures.

3.1.5.1 (14) Intersystem standard compliance

Problem	Type & Action	Suggestion
Weak coherence between "Metric Name", "Purpose of the metrics" and external measure "Interface standard compliance"	SM, PC Mod	Change the "Metric Name" to "System interface standard compliance".

3.2 Reliability measures

3.2.1 Maturity measures.

3.2.1.1 (15) Fault detection

Problem	Type & Action	Suggestion
Clarify the source reference data.	AC Mod	Add the comment "This measure must only be used if the past history or a reference model exit".
Weak coherence between "Purpose of the metrics" and "Measurement, formula"	SM Mod	Change the "Purpose of the metrics" to "How many estimated faults were detected in reviewed product?"
Weak coherence between "Measurement, formula", and "Interpretation of value"	SM Mod	Change the "Interpretation of value" to "[…] A high value for X implies a high number of faults detected".
Hidden information	AC Mod	Add to the "Measurement, formula": "X=A A=Number of faults detected". Add to "Interpretation of value" 0<=X A high value of X implies bad product quality".

3.2.1.2 (16) Fault density

Problem	Type & Action	Suggestion
No measure for fault detection with real data, and a prediction for external measure "Fault density".	AC, PC, PT Add	Add the measure "Fault density"
• Purpose of the measure:		How many faults were detected in reviewed product?
• Method of application:		Count the number of detected faults in evaluation and compute density.
• Suggested Technical quality measure:		Size sub-characteristic. OO Size sub-characteristic.

3.2.1.3 (17) Test adequacy

Problem	Type & Action	Suggestion
Meaning of "test case" in "Purpose of metrics"	SM Mod	Add the comment: "Test case: a set of test inputs, execution conditions and expected results developed for a particular objective."

3.2.2 Fault Tolerance measures.

3.2.2.1 (18) Failure avoidance

Problem	Type & Action	Suggestion
Extend the nature of "failure pattern"	PT Mod	Add the comment "Fault pattern may be affected to software product complexity".
Relationship with software product complexity.	PT Mod	Add the column: "Suggested Technical quality measure: Complexity and OO Complexity sub-characteristic."

3.2.2.2 (19) Incorrect operation avoidance

Problem	Type & Action	Suggestion
Duplicated purpose of the measure.	AC Del	There is a big similarity between "fault" and "incorrect operation", the objective of the deleted measure may be covered by "Failure avoidance" (an "incorrect operation" can be a "fault pattern"). In other sense, which other fault patterns must be measured in this specific measurement?

3.2.3 Recoverability measures.

3.2.3.1 (20) Restoration effectiveness

Problem	Type & Action	Suggestion
It need documented sources to calculate restoration time.	PT Mod	As an option in calculation restoration time, it is possible to analyze restoration algorithm's complexity and related it to time behavior [RAB99]. So add the column: "Suggested Technical quality measure: Complexity sub-characteristic. OO Complexity sub-characteristic."

3.3 Usability measures.

3.3.1 Understandability measures.

3.3.1.1 (21) Demonstration capability

Problem	Type & Action	Suggestion
Accurate use of "demonstration" concept with external measure "Demonstration capability"	PC, SM Mod	Change in all columns the text "demonstration" to "demonstration / tutorial".

3.3.1.2 (22) Demonstration consistency

Problem	Type & Action	Suggestion
No measure for tutorial consistency and a prediction for external measure "Demonstration accessibility" and "Demonstration effectiveness".	AC, PC Add	Add the measure "Demonstration consistency"
• Purpose of the measure:		How many functions that are demonstrable behave the same way to similar demonstrable functions in other parts of the system?
• Method of application:		Count the number of demonstrable functions with inconsistency behavior and compare it with the total of demonstrable functions specified in requirements.

3.3.1.3 (23) User Interface consistency

Problem	Type & Action	Suggestion
No measure for user interface consistency and a prediction for external measures "Demonstration accessibility" and "Demonstration effectiveness".	AC, PC Add	Add the measure "User Interface consistency"
• Purpose of the measure:		How many user interfaces behave the same way to similar user interfaces in other parts of the system?
• Method of application:		Count the number of user interfaces with inconsistency behavior and compare it with the total of user interfaces specified in requirements and/or design.

3.3.1.4 (24) User Documentation consistency

Problem	Type & Action	Suggestion
No measure for user documentation consistency and a prediction for external measures "Demonstration accessibility", "Demonstration effectiveness", "Understandable input and output" and "Ease of function learning"	AC, PC Add	Add the measure "User Documentation consistency"
• Purpose of the measure:		How many user documents are structured in the same way to similar user documents in the system?
• Method of application:		Count the number of user documents with inconsistency structure and compare it with the total of user documents specified in requirements.

3.3.1.5 (25) Output Messages consistency

Problem	Type & Action	Suggestion
No measure for user documentation consistency and a prediction for external measures "Demonstration accessibility", "Demonstration effectiveness", "Understandable input and output" and "Ease of function learning"	AC, PC Add	Add the measure "Output Messages consistency"
• Purpose of the measure:		How many output messages are structured in the same way to similar output messages in the system?

• Method of application:		Count the number of output messages with inconsistency structure and compare it with the total of output messages specified in requirements and/or design.

3.3.2 Learnability measures.

3.3.2.1 (26) Help consistency

Problem	Type & Action	Suggestion
No measure for help consistency and a prediction for external measure "Help accessibility" and "Effectiveness of help systems".	AC, PT, PC Add	Add the measure "Help consistency"
• Purpose of the measure:		What proportion of help topics behaves the same way to similar help topics in other parts of the system?
• Method of application:		Count the number of help topics with inconsistency behavior and compare it with the total of help topics specified in requirements and/or design.

3.3.2.2 (27) Help understandability

Problem	Type & Action	Suggestion
No measure for help understandability and a prediction for external measure "Help accessibility", "Effectiveness of help systems" and "Help frequency".	AC, PC Add	Add the measure "Help understandability"
• Purpose of the measure:		What proportion of help topics the user will be able to correctly understand?
• Method of application:		Count the number of help topics with incomprehensive information (in a peer review for example) and compare it with the total of help topics specified in requirements and/or design.

3.3.2.3 (28) Completeness of Training material

Problem	Type & Action	Suggestion
No measure for Completeness of training material, and a prediction for external measure "Ease of function learning".	AC, PC Add	Add the measure "Completeness of training material"
• Purpose of the measure:		What proportion of functions is described in the training material?
• Method of application:		Count the number of implemented functions whit training material and compare with the total number of functions as specified in requirements.

3.3.3 Operability measures.

3.3.3.1 (29) Default value availability

Problem	Type & Action	Suggestion
No measure for default value availability and a prediction for external measure "Default value availability in use".	AC, PC Add	Add the measure "Default value availability"
• Purpose of the measure:		What proportion of functions and/or input items have default value availability?
• Method of application:		Count the number of implemented functions and/or input items whit default values availability as specified in requirements and/or design.

3.3.3.2 (30) Operation procedure reduction availability

Problem	Type & Action	Suggestion
No measure for Operation procedure reduction and a prediction for external measure "Operation procedure reduction".	AC, PC Add	Add the measure "Operation procedure reduction"
• Purpose of the measure:		What proportion of functions will permit the user to reduce operation procedures?
• Method of application:		Count the number of implemented functions whit operation procedure reduction availability as specified in requirements and/or design.

3.3.4 Compliance for Usability measures.

3.3.4.1 (31) Consistency with other systems

Problem	Type & Action	Suggestion
No measure for Consistency with others known systems.	AC, PC Add	Add the measure "Consistency with others known systems"
• Purpose of the measure:		How consistent is the product in relation to others known systems?
• Method of application:		Count the number of items requiring consistency with other knows systems and compare with the number of items requiring consistency with others know systems as in the specifications and/or design.

3.4 Efficiency measures.

3.4.1 Time behavior measures.

3.4.1.1 (32) Response time

Problem	Type & Action	Suggestion
Weak coherence between "Metric Name", "Purpose of the metrics" and "Method of application"	AC, PC, PT Mod	Change "Metric name" to "Specified task response time".
Meaning of "task"	SM Mod	Add the comment: "Task: an elementary process (input, input transformation and input transformed output), with a well defined purpose."
Unrealistic or not well described "Method of application".	AC, PC, PT Mod	Add the comments: "It is possible to estimate task time based in task algorithm complexity analysis. It is a MIS systems result interpretation."
• Method of application:	AC, PC, PT Mod	Define the limits of the specified task. Calculate the task complexity, estimate the elements involve in task description as defined in requirements and design. Estimate the response time in function of the size of data (n).
• Measurement, formula:	AC, PC, PT Mod	X= Estimated software platform response time (n) + Estimated data management response time (n) + Estimated transmission response time (n) + algorithms set complexity of the specified task (n).
• Suggested technical measures:	PT Mod	Complexity. OO Complexity.

3.4.1.2 (33) Estimated Software platform response time

Problem	Type & Action	Suggestion
No measure to estimate the software platform response time.	AC, PC Add	Add the measure "Estimated Software platform response time"
		Comment: "Software platform = operating system, application server, transaction monitor, call center system, etc". Comment: "Software platform performance data = response time described by the provider or technical sheets for software platform". Comment: "It is a MIS systems result interpretation."
• Purpose of the measure:		What is the estimated response time of software platform needed to complete a specified task?
• Method of application:		Use the software platform performance data (bytes/time) and the result of measure "Estimated I/O Utilization size".
• Measurement, formula:		X= Sum (response time of software platform's element) * "Estimated I/O Utilization size" of specified task.
• Interpretation of measure:		The shorter the better.

3.4.1.3 (34) Estimated Data management response time

Problem	Type & Action	Suggestion
No measure to estimate the Data management response time.	AC, PC Add	Add the measure "Estimated Data management response time"
		Comment: "Data management = data base management systems, file manager systems, searching systems, expert systems, etc." Comment: "Data management performance data = response time described by the provider or technical sheets for data management systems." Comment: "It is a MIS systems result interpretation."
• Purpose of the measure:		What is the estimated response time of data management needed to complete a specified task?

• Method of application:		Use the data management performance data (bytes/time) and the result of measure "Estimated I/O Utilization size".
• Measurement, formula:		X= Sum (response time of data management's element) * "Estimated I/O Utilization size" of specified task.
• Interpretation of measure:		The shorter the better.

3.4.1.4 (35) Estimated Transmission response time

Problem	Type & Action	Suggestion
No measure to estimate the Transmission response time.	AC, PC Add	Add the measure "Estimated Transmission response time"
		Comment: "Transmission = intranet, internet, local networks, wireless communications, etc." Comment: "Transmission performance data = response time described by the provider or technical sheets for transmission systems". Comment: "It is a MIS systems result interpretation."
• Purpose of the measure:		What is the estimated response time of transmission needed to complete a specified task?
• Method of application:		Use the transmission performance data (bytes/time) and the result of measure "Estimated I/O Utilization size".
• Measurement, formula:		X= Sum (response time of transmission's element) * "Estimated I/O Utilization size" of specified task.
• Interpretation of measure:		The shorter the better.

3.4.1.5 (36) Throughput time

Problem	Type & Action	Suggestion
Meaning of "task"	SM Mod	Add the comment: "Task: an elementary process (input, input transformation and input transformed output), with a well defined purpose."
Unrealistic or not well described "Method of application".	AC, PC, PT Mod	Add the comments: "Software platform performance data = response time described by the provider or technical sheets for operating system, application server, transaction monitor, call center system, etc. Data management software performance data = response time described by the provider or technical sheets for data base management systems, file manager systems, searching systems, expert systems, etc. Transmission software and hardware performance data = response time described by the provider or technical sheets for intranet, internet, local networks, unwired communications, etc. It is possible to estimate task time based in task algorithm complexity analysis. It is a MIS systems result interpretation."
• Purpose of the measure:	AC, PC, PT Mod	What is the estimated number of same-type tasks that can be performed over a unit of time?
• Method of application:	AC, PC, PT Mod	Classify tasks according their functionality. Use software platform performance data, data management performance data, transmission performance data and task complexity data to estimate the response time of a specific task type. Estimate the response time using one data unit size.
• Measurement, formula:	AC, PC, PT Mod	Y= Fixed response time of all software platform's elements (1 unit of data size) + fixed response time of all data management's elements (1 unit of data size) + fixed response time of all transmission's elements (unit of data size) + the algorithms set of a specified task type complexity (1 unit of data size) X = unit time / Y
• Suggested technical measures:	PT Mod	Complexity. OO Complexity.

3.4.1.6 (37) Turnaround time

Problem	Type & Action	Suggestion
Duplicated purpose of the measure.	AC Del	We can use the "Specified task response time" measure to measure a group of tasks (according to the algorithm used for complexity analysis). Or we can define a new specified task that encloses a set of tasks. The estimated response time is a function of data size (n).

3.4.2 Resource Utilization measures.

3.4.2.1 (38) I/O Utilization

Problem	Type & Action	Suggestion
Weak coherence between "Metric Name", "Purpose of the metrics" and "Method of application"	AC, PT Mod	Change "Metric name" to "Estimated I/O Utilization". Change "Method of application" to "[…] for the task." Change "Measure type" to "X=count".
Meaning of "task"	SM Mod	Add the comment: "Task: an elementary process (input, input transformation and input transformed output), with a well defined purpose."
• Method of application:	AC, PT Mod	Count total number of task input and output events as defined in requirements, design and code.
• Measurement, formula:	AC, PT Mod	X= Total number of individual readings events defined of coded for a specified task (inputs) + Total number of individual writings events defined of coded for a specified task (outputs).

3.4.2.2 (39) Estimated I/O Utilization size

Problem	Type & Action	Suggestion
No measure for I/O buffering, and a prediction for external measure "I/O loaded limits".	AC, PC Add	Add the measure "Estimated I/O Utilization size"
• Purpose of the measure:		What is the estimated I/O utilization size (buffering) in a specified task?
• Method of application:		Count the maxim size of task data input and data output (according data type definition) as defined in requirements, design and code.
• Measurement, formula:		Y= Total input data length of specified task (input buffer size) + Total output length data size of specified task (output buffer size).

3.4.2.3 (40) I/O Utilization Message Density

Problem	Type & Action	Suggestion
Weak coherence between "Metric Name" and "Method of application". This is not an estimated measure.	AC, PT Mod	Change "Method of application" to "[…] compare it to number of lines […]"

3.4.2.4 (41) Memory Utilization Message Density

Problem	Type & Action	Suggestion
Weak coherence between "Metric Name" and "Method of application". This is not a estimated measure.	AC, PT Mod	Change "Method of application" to "[…] compare it to number of lines […]"

3.4.2.5 (42) Transmission time

Problem	Type & Action	Suggestion
Duplicated purpose of the measure.	AC Del	The "Estimated Transmission response time", "I/O Utilization" and "Estimated I/O Utilization size" measures could cover the effects of the "Transmission time" measure.

3.5 Maintainability measures.

3.5.1 Analyzability measures.

3.5.1.1 (43) Activity recording legibility

Problem	Type & Action	Suggestion
No measure for activity recording legibility and a prediction for external measure "Failure analysis efficiency" and "Status monitoring capability". To avoid recording unnecessary information.	AC, PC Add	Add the measure "Activity recording legibility"
• Purpose of the measure:		How useful and complete is the information recorded in the activity log?
• Method of application:		Count the number of duplicated, unnecessary or incomplete items logged in the activity log as specified and compare it to the number of items that are required to be logged.
• Measurement, formula:		X=1-A/B A=Number of duplicated, unnecessary or incomplete implemented data login items confirmed in evaluation. B=Number of data items to be logged defined in the specifications.

3.5.2 Changeability measures.

3.5.2.1 (44) Modification complexity

Problem	Type & Action	Suggestion
No measure for modification complexity and a prediction for external measure "Modification complexity" and "Change implementation elapse time".	AC, PC, PT Add	Add the measure "Modification complexity"
• Purpose of the measure:		Which is the complexity in function modification?
• Method of application:		Calculate the functions complexity according to a predefined scale.
• Measurement, formula:		X= Complexity (function) Use the technical quality measures to calculate the function complexity.
• Measure scale type:		Ordinal
• Measure type:		X= Score
• Suggested technique measures:		Complexity sub-characteristic. OO Complexity sub-characteristic.
• Interpretation of measured:		The lesser the better.

3.5.2.2 (45) Parameterized availability

Problem	Type & Action	Suggestion
No measure for parameterized availability and for the prediction of the external measure "Parameterized modifiability".	AC, PC Add	Add the measure "Parameterized availability"
• Purpose of the measure:		Which is the reusability level of implemented functions?
• Method of application:		Count the number of functions implemented with parameterized availability and compare with the total number of functions specified.
• Measurement, formula:		X= A/B A=Number of functions with parameterized availability. B=Total of number of functions specified.

3.5.2.3 (46) Reuse utilization

Problem	Type & Action	Suggestion
No way to know reuse utilization.	AC Add	Add the measure "Reuse utilization"
• Purpose of the measure:		What proportion of functions could be reutilized?
• Method of application:		Calculate reusability level of implemented functions according to a predefined scale.
• Measurement, formula:		X= Reusability (function) Use the technical quality measures to calculate the function reusability.
• Measure scale type:		Ordinal
• Measure type:		X= Score
• Suggested technique measures:		Self-descriptiveness sub-characteristic. OO Maintainability & reusability sub-characteristic. OO Reusability sub-characteristic.
• Interpretation of measured:		The greater the better.

3.5.2.4 (47) Programming Style consistency

Problem	Type & Action	Suggestion
No measure for style programming consistency and for prediction of the external measures "Modification complexity" and "Change implementation elapse time".	AC, PC Add	Add the measure "Programming style consistency"
• Purpose of the measure:		How many functions are implemented using the same programming style or programming standards?
• Method of application:		Count the number of functions implemented using the same programming style or programming standards and compare it with the total of functions specified in requirements and/or design.
• Measurement, formula:		X= A/B A=Number of functions implemented using the same programming style or programming standards. B=Total of number of functions specified.

3.5.2.5 (48) Frameworks utilization

Problem	Type & Action	Suggestion
No framework utilization measure from a reusability perspective.	AC Add	Add the measure "Frameworks utilization"
• Purpose of the measure:		What proportion of functions specified is part of a framework?
• Method of application:		Count the number of functions that are founded in a know framework and compare it with the total of functions specified in requirements and/or design.
• Measurement, formula:		X= A/B A=Number of functions in a framework as view in evaluation. B=Total of number of functions specified.
• Interpretation of measured:		The greater the better.

3.5.2.6 (49) Patterns utilization

Problem	Type & Action	Suggestion
No patterns utilization measure from a reusability perspective.	AC Add	Add the measure "Patterns utilization"
• Purpose of the measure:		What proportion of functions specified is part of a pattern?
• Method of application:		Count the number of functions that are founded in a know patterns and compare it with the total of functions specified in requirements and/or design.
• Measurement, formula:		X= A/B A=Number of functions in a patterns as view in evaluation. B=Total of number of functions specified.
• Interpretation of measured:		The greater the better.

3.5.2.7 (50) Programs libraries utilization

Problem	Type & Action	Suggestion
No program libraries use measure.	AC Add	Add the measure "Programs libraries utilization"
• Purpose of the measure:		What proportion of functions use a know programs library?
• Method of application:		Count the number of functions that use a know programs library and compare it with the total of functions specified in requirements and/or design.
• Measurement, formula:		X= A/B A=Number of functions that use a know programs library as view in evaluation. B=Total of number of functions specified.

3.5.2.8 (51) Data stores / procedures utilization

1Problem	Type & Action	Suggestion
No measure of data stores and data procedures use.	AC Add	Add the measure "Data stores / procedures utilization"
• Purpose of the measure:		What proportion of functions use an existed data stores or data procedures?
• Method of application:		Count the number of functions that use an existed data stores or data procedures and compare it with the total of functions specified in requirements and/or design.
• Measurement, formula:		X= A/B A=Number of functions that use an existed data stores or data procedures as view in evaluation. B=Total of number of functions specified.

3.5.3 Stability measures.

3.5.3.1 (52) Technical Documentation consistency

Problem	Type & Action	Suggestion
No technical documentation consistency measure.	AC Add	Add the measure "Technical Documentation consistency"
• Purpose of the measure:		How many technical documents are structured in the same way to similar technical documents in the system?
• Method of application:		Count the number of technical documents with inconsistency structure and compare it with the total of technical documents specified in requirements.

3.5.3.2 (53) Technical Documentation understandability

Problem	Type & Action	Suggestion
No technical documentation understandability measure.	AC Add	Add the measure "Technical documentation understandability
• Purpose of the measure:		What proportion of help topics will the software developer team be able to understand correctly?
• Method of application:		Count the number of technical documentation with incomprehensive information (in a peer review for example) and compare it with the total of technical documentation specified in requirements and/or design.

3.5.3.3 (54) Completeness of Technical Documentation

Problem	Type & Action	Suggestion
No completeness of technical documentation measure.	AC Add	Add the measure "Completeness of technical documentation"
• Purpose of the measure:		What proportion of functions is described in the technical documentation?
• Method of application:		Count the number of implemented functions whit technical documentation and compare with the total number of functions as specified in requirements.

3.6 Portability measures.

3.6.1 Adaptability measures.

3.6.1.1 (55) Internationalization

Problem	Type & Action	Suggestion
No internationalization portability measure.	AC Add	Add the measure "Internationalization"
• Purpose of the measure:		How adaptable is the product to internationalization aspects?
• Method of application:		Count the number of implemented functions that are capable of achieving required results in specified countries and compare it to the number of functions with internationalization capability requirements.
• Measurement, formula:		X= A/B A=Number of functions that are capable of achieving internationalization as view in evaluation. B=Total of number of functions specified.

3.6.2 Installability measures.

3.6.2.1 (56) Easy of installation

Problem	Type & Action	Suggestion
No complexity of installation measure and a prediction for external measure "Easy of installation".	AC, PC Add	Add the measure "Easy of installation"
• Purpose of the measure:		How easy is it to install the product?
• Method of application:		Count the number of complex implemented setup operations and compare it to the number of setup operation specified.
• Measurement, formula:		X= 1-A/B A=Number of functions that have complex implemented setup operations as view in evaluation. B=Total of number of setup operations specified.

4. IMPROVEMENT OPTIONS

All the hypotheses and related research discussed in previous clauses have rendered results that could prove useful for software engineering community, with a special emphasis on the software quality engineering part. The possible recommendations of changes go into two basic categories:

- Changes to ISO/IEC 9126 quality model, and
- Interventions to internal quality measures taking form of modifications or deletions of existing measures and additions of new measures.

Recommendations of changes to the quality model seem to be at the first glance particularly inconvenient or even dangerous. The ISO/IEC 9126 model for internal and external quality is present in the industry and research for over 13 years gaining during this time the unquestionable international recognition. From this perspective almost any change to the model could meet mixed reactions from the international community of users; however, a quick analysis of the model proposed in this paper will reveal that the central part of the model being the core ISO/IEC 9126 quality model remains intact while the modifications appear as side enhancements. This particular form of possible changes has been proposed with two objectives in mind:

- Keeping the commonality feature of the model intact (e.g. the core of the model is still common for internal and external quality),
- Keeping the original core of the model intact so the actual users may still reference it as before if they do not wish to use the enhanced part of the overall model.

Such an approach allows then for a full implementation of changes to the model recommended in this paper if the editors and experts from ISO (Subcommittee SC7 – Systems and software engineering, Working Group WG6) agree upon it. If this is the case, the next section presents the suggested ISO/IEC 9126 modified measures tables.

5. Suggested Internal Quality Measures Tables

Functionality measures.

Functionality. Suitability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
1 1[1]	**Functional adequacy**	How adequate are the checked functions?	Count the number of implemented functions that are suitable for performing the specified tasks, and then measure the ratio of it to functions implemented.	X=1-A/B A= Number of functions in which problems are detected in evaluation B= Number of functions checked	0 <= X <= 1 The closer to 1, the more adequate.	Absolute	X=count/count A=count B=count	
• Synonymous of adequacy: satisfactory or acceptable. • Function or functional item may be a Functional Requirement, a Module, a Procedure, a programmable Function, a Stored Procedure, a Use Case, a Class, a Class Method, an Object or an Object Method, descriptions and source code (it is not an exhaustive list).								
2 2	**Functional implementation completeness**	How complete is the functional implementation?	Count the number of missing functions detected in evaluation and compare with the number of function described in the requirement specifications	X=1-A/B A=Number of missing functions detected in evaluation. B=Number of functions described in requirement specifications	0 <= X <= 1 The closer to 1, the more complete.	Absolute	X=count/count A=count B=count	
• Synonymous of completeness are: having all the necessary parts or entire. • Implementation or a function item must be implemented with a Function Model, Data Model or Source Code.								
3 3	**Functional implementation correctness**	How correct is the functional implementation?	Count the number of incorrectly implemented functions and compare with the number of functions described in the requirement specifications	X=1-A/B A= Number of incorrectly implemented or missing functions detected. B= Number of functions described in requirement specifications	0 <= X <= 1 The closer to 1, the more correct.	Absolute	X=count/count A=count B=count	
• Synonymous of correctness are: free from error or true or right.								
4 4	**Functional specification stability (volatility)**	How stable is the functional specification during the development life cycle?	Count the number of functions changed (added, modified, or deleted) during development life cycle phase, then compare with the number of functions described in the requirement specifications.}	X=1-A/B A=Number of functions changed during development life cycle phases B=Number of functions described in requirement specifications	0 <= X <= 1 The closer to 1 the more stable.	Absolute	X=Count/ Count A=Count B=Count	

Functionality. Accuracy measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
5 5	**Computational accuracy**	How correct have the mathematical calculation been implemented?	Count the number of mathematical calculations that have been correctly implemented and compare with the number of mathematical calculations described in requirement specifications.	X=A/B A=Number of accurate mathematical calculations as confirmed in evaluation. B=Number of mathematical calculations described in requirement specifications.	0<=X<=1. The closer to 1, the more accurate.	Absolute	X=count/count A=count B=count	

[1] *The red color numeration corresponds to the list of 56 modifications made to the ISO/IEC 9126-3 measures described in section 4 of this paper.*

- Computational synonymous: mathematical calculation, accuracy synonymous: correct in all details

5b 6	**Computational completeness**	How complete have the mathematical calculations been implemented?	Count the number of mathematical calculations that have been implemented and compare with the number of mathematical calculations described in requirement specifications.	X=1-A/B A=Number of missing mathematical calculations detected in evaluation. B=Number of mathematical calculations described in requirement specifications.	0<=X<=1. The closer to 1, the more complete.	Absolute	X=count/count A=count B=count
6 7	**Precision completeness**	How complete was the implementation of specific levels of precision for the data items?	Count the number of data items that meet the requirements of specific levels of precision and compare to the total number of data items with specific level of precision requirements.	X=A/B A= Number of data items implemented with specific levels of precision, confirmed in evaluation B= Number of data items that require specific levels of precision	0 <= X <= 1. The closer to 1, the more complete.	Absolute	X=count/count A=count B=count

- Data item (data): an elementary piece of information. Precision: valid range of data type units. Data type examples: integer, real, date, string, etc

6a 8	**Precision accuracy**	How correct have the precision of data items been implemented?	Count the number of data items that have been implemented with the incorrect level of precision and compare with the total number of data items with specific level of precision described in requirement specifications.	X=1-A/B A=Number of data items implemented with incorrect level of precision detected in evaluation. B=Number of data items that require specific level of precision.	0<=X<=1. The closer to 1, the more accurate.	Absolute	X=count/count A=count B=count

Functionality. Interoperability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
7 9	**Data exchangeability accuracy (data format based)**	How correct have the data formats used as interface with other software or system been implemented?	Count the number of interface data formats that have been implemented correctly as in the specifications and compare to the number of data formats to be exchanged as in the specifications	X=A/B A=Number of interface data formats that have been implemented correctly as in the specifications B=Number of data formats to be exchanged as in the specifications	0 <= X <= 1. The closer to 1, the more correct.	Absolute	X=count/count A=count B=count	
7a 10	Data exchangeability completeness (data format based)	How complete have the data formats used as interface with other software or system been implemented?	Count the number of data formats used as interface that have been implemented and compare with the number of data formats used as interface described in requirement specifications.	X=1-A/B A=Number of missing data formats used as interface detected in evaluation. B=Number of data formats used as interface described in requirement specifications.	0<=X<=1. The closer to 1, the more complete.	Absolute	X=count/count A=count B=count	
8 11	**Interface protocol consistency**	How correct, complete and unchanging have the interface protocols been implemented?	Count the number of interface protocols that were implemented correctly, complete and unchanged as in the specifications and compare with the number of interface protocols to be implemented as in the specifications.	X=A/B A=Number of interface protocols that have been implemented correctly, complete and unchanged as detected in evaluation. B=Number of interface protocols to be implemented as in the specifications	0 <= X <= 1 The closer to 1, the more consistent.	Absolute	X=count/count A=count B=count	

- Consistency synonymous: conforming to a regular pattern or unchanging

Functionality. Security measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
9 12	**Access auditability**	How auditable is access to the system?	Count the number of access types that are being logged correctly as in the specifications and compare with the	X=A/B A= Number of access types that are being recorded as in the specifications	0 <= X <= 1 The closer to 1, the more auditable.	Absolute	X=count/count A=count B=count	

			number of access types that are required to be recorded as described in the specifications.	B= Number of access types required to be recorded in the specifications				

- Login is one of the access types to be recorded

10	**Access controllability**	How controllable is access to the system?	Count the number of access controllability requirements implemented correctly as in the specifications and compare with the number of access controllability requirements in the specifications.	X=A/B A= Number of access controllability requirements implemented correctly as in the specifications. B= Number of access controllability requirements in the specifications..	0 <= X <= 1 The closer to 1, the more controllable.	Absolute	X=count/count A=count B=count	
11 13	**Data corruption prevention**	How complete is the implementation of data corruption prevention?	Count the number of implemented instances of data corruption prevention as specified and compare with the number of illegal operations specified in requirements as capable of corrupting/ destroying data.	X=A/B A= Number of implemented instances of data corruption prevention as specified confirmed in review. B= Number of illegal operation specified in requirements.	0 <= X <= 1 The closer to 1, the more preventive.	Absolute	X=count/count A=count B=count	

- Illegal operations: operations capable of corrupting or destroying data

12	**Data encryption**	How complete is the implementation of data encryption?	Count the number of implemented instances of encryptable/decryptab le data items as specified and compare with the number of instances of data items requiring data encryption/decryption facility as in specifications.	X=A/B A=Number of implemented instances of encryptable/decryptable data items as specified confirmed in review B= Number of data items requiring data encryption/decryption facility as in specifications	0 <= X <= 1 The closer to 1, the more complete.	Absolute	X=Count/ Count A=Count B=Count	

Functionality. Compliance measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
13	**Functional Compliance**	How compliant is the functionality of the product to applicable regulations, standards and conventions.	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.	X=A/B A= Number of correctly implemented items related to functionality compliance confirmed in evaluation B= Total number of compliance items	0 <= X <= 1. The closer to 1, the more compliant.	Absolute	X=count/count A=count B=count	
14 14	**System interface standard compliance**	How compliant are the interfaces to applicable regulations, standards and conventions	Count the number of interfaces that meet required compliance and compare with the number of interfaces requiring compliance as in the specifications	X=A/B A= Number of correctly implemented interfaces as specified, confirmed in review B= Total number of interfaces requiring compliance	0 <= X <= 1. The closer to 1, the more compliant.	Absolute	X=count/count A=count B=count	

Reliability measures.

Reliability. Maturity measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
15 15	**Fault detection**	How many estimated faults were detected in reviewed product?	Count the number of detected faults in review and compare it to the number of estimated faults to be detected in this phase.	X=A A=Number of faults detected.	0 <=X A high value of X implies bad product quality	Absolute	X=count A=count	
				X=A/B A=Absolute number of faults detected in review B=Number of estimated faults to be detected in review (using past history or reference model)	0 <= X A high value for X implies high number of faults detected.	Absolute	X=count/count A=count B=count	

- This measure must only be used if it exits past history or a reference model.
- It is necessary to convert this value (X) to the <0,1> interval if making summarization of characteristics. Value A comes from review report Value B comes from the organization database.

15a 16	**Fault density**	How many faults were detected in reviewed product?	Count the number of detected faults in evaluation and compute density.	X=A/B A=Number of detected faults in evaluation. B=Product size.	It depends on stage of testing. At early stage, larger is better. At the later stages, smaller is better. It is necessary to convert this value (X) to the <0,1> interval if making summarization of characteristics.	Absolute	X=count/size. A=count B=count	Size. OO Size.
16	**Fault removal**	How many faults have been corrected? What is the proportion of faults removed?	Count the number of faults removed during design/coding and compare it to the number of faults detected in review during design/coding.	X=A A=Number of corrected faults in design/coding Y=A/B A=Number of corrected faults design/coding B= Number of faults detected in review	0 <= X A high value of X implies, that fewer faults remain. 0 <= Y <= 1 The closer to 1, the better (more faults removed)	Ratio Absolute	X=count/count A=count B=count	

- It is necessary to convert this value (X) to the <0,1> interval if making summarization of characteristics. Value A comes from review report Value B comes from the organization database.

17 17	**Test adequacy**	How much of the required test cases is covered by the test plan?	Count the number of test cases planned and compare it to the number of test cases required to obtain adequate test coverage.	X=A/B A=Number of test cases designed in test plan and confirmed in review B= Number of test cases required	0 <= X Where X is greater the better adequacy	Absolute	X=count/count A=count B=count	

- Test case: a set of test inputs, execution conditions and expected results developed for a particular objective.

Reliability. Fault Tolerance measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
18	**Failure avoidance**	How many fault	Count the number of avoided fault	X=A/B	0 <= X	Absolute	X=count/count	Complexity.

18		patterns were brought under control to avoid critical and serious failures?	patterns and compare it to the number of fault patterns to be considered	A=Number of fault patterns having avoidance in design/code B=Number of fault patterns to be considered	Where X is greater the better failure avoidance		A=count B=count	OO Complexity.

- Fault pattern may be affected by software product complexity.
- Fault pattern examples are: out of range and data deadlock.
- Fault tree analysis technique may be used to detect fault patterns.

19

Reliability. Recoverability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
20	**Restorability**	How capable is the product in restoring itself after abnormal event or at request?	Count the number of implemented restoration requirements and compare it to the number of restoration requirements in the specifications. Restoration requirement examples: database checkpoint, Transaction checkpoint, redo function, undo function	X=A/B A=Number of implemented restoration requirements confirmed in review B=Number of restoration requirements in the specifications.	0 <= X <= 1 Where X is greater, the better restorability	Absolute	X=count/count A=count B=count	
21 20	**Restoration Effectiveness**	How effective is the restoration capability?	Count the number of implemented restoration requirements meting target restoration time (by calculations or simulations) and compare it to the number of restoration requirements with specified target time.	X=A/B A=Number of implemented restoration requirements meeting target restore time B=Number of restoration requirements with specified target times	0 <= X <= 1 Where X is greater, the better effectiveness	Absolute	X=count/count A=count B=count	Complexity. OO Complexity.

Reliability. Compliance measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
22	**Reliability Compliance**	How compliant is the reliability of the product to applicable regulations, standards and conventions.	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.	X=A/B A= Number of correctly implemented items related to reliability compliance confirmed in evaluation B= Total number of compliance items	0 <= X <= 1. The closer to 1, the more compliant.	Absolute	X=count/count A=count B=count	

Usability measures.

Usability. Understandability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
23	**Completeness of description**	What proportion of functions (or types of function) is described in the product description?	Count the number of functions that are adequately described and compare with the total number of functions in the product.	X= A/B A= Number of functions (or types of functions) described in the product description B= Total number of functions (or types of functions)	0<=X<=1 The closer to 1 the more complete	Absolute	X=count/count A=count B=count	
	• This indicates whether potential users will understand the capability of the product after reading the product description. • See also ISO/IEC 9127 Consumer software packages.							
24 21	**Demonstration capability**	What proportion of functions requiring demonstration / tutorial has demonstration capability?	Count the number of functions that are adequately demonstrable and compare with the total number of functions requiring demonstration / tutorial capability	X=A/B A= Number of functions demonstrated and confirmed in review B= Total number of functions requiring demonstration capability	0<=X<=1 The closer to 1 the more capable.	Absolute	X=count/count A=count B=count	
	• Demonstrations step through the process showing how the product is used. This includes "wizards".							
24a 22	**Demonstration consistency**	How many functions that are demonstrable behave the same way to similar demonstrable functions in other parts of the system?	Count the number of demonstrable functions with inconsistency behavior and compare it with the total of demonstrable functions specified in requirements.	X=1-A/B A= Number of demonstrable functions with inconsistency behavior. B= Total of demonstrable functions specified in requirements and/or design.	0<=X<=1 The closer to 1, the more consistent.	Absolute	X=count/count A=count B=count	
25	**Evident functions**	What proportion of the product functions is evident to the user?	Count the number of functions that are evident to the user and compare with the total number of functions	X= A/B A= Number of functions (or types of functions) evident to the user B= Total number of functions (or types of functions)	0<=X<=1 The closer to 1 the better	Absolute	X=count/count A=count B=count	
	• This indicates whether users will be able to locate functions by exploring the interface (e.g. by inspecting the menus)							
25a 23	**User Interface consistency**	How many user interfaces behave the same way to similar user interfaces in other parts of the system?	Count the number of user interfaces with inconsistency behavior and compare it with the total of user interfaces specified in requirements and/or design.	X=1-A/B A= Number of user interfaces with inconsistency behavior. B= Total of user interfaces specified in requirements and/or design.	0<=X<=1 The closer to 1, the more consistent.	Absolute	X=count/count A=count B=count	
26	**Function understandability**	What proportion of the product functions will the user be able to understand correctly?	Count the number of user interface functions where purposes are understood by the user and compare with the number of user interface functions.	X= A/B A= Number of user interface functions whose purpose is understood by the user B= Number of user interface functions.	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	
26a 24	**User Documentation consistency**	How many user documents are structured in the same way to similar user documents in the system?	Count the number of user documents with inconsistency structure and compare it with the total of user documents specified in requirements.	X=1-A/B A= Number of user documents with inconsistency structure. B= Total of user documents specified in requirements.	0<=X<=1 The closer to 1, the more consistent.	Absolute	X=count/count A=count B=count	

26b 25	**Output Messages consistency**	How many output messages are structured in the same way to similar output messages in the system?	Count the number of output messages with inconsistency structure and compare it with the total of output messages specified in requirements and/or design.	X=1-A/B A= Number of output messages with inconsistency structure. B= Total of output messages specified in requirements and/or design.	0<=X<=1 The closer to 1, the more consistent.	Absolute	X=count/count A=count B=count	

Usability. Learnability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
27	**Completeness of user documentation and/or help facility**	What proportion of functions is described in the user documentation and/or help facility?	Count the number of functions implemented with help facility and/or documentation and compare with the total number of functions in product.	X= A/B A= Number of functions described B= Total of number of functions provided	0 <= X <= 1 The closer to 1,	Absolute	X=count/count A=count B=count	

- Three metrics are possible: completeness of the documentation, completeness of the help facility or completeness of the help and documentation used in combination.

27a 26	**Help consistency**	What proportion of help topics behaves the same way to similar help topics in other parts of the system?	Count the number of help topics with inconsistency behavior and compare it with the total of help topics specified in requirements and/or design.	X=1-A/B A= Number of help topics with inconsistency behavior. B= Total of help topics specified in requirements and/or design.	0<=X<=1 The closer to 1, the more consistent.	Absolute	X=count/count A=count B=count	
27b 27	**Help understandability**	What proportion of help topics the user will able to correctly understand?	Count the number of help topics with incomprehensive information (in a peer review for example) and compare it with the total of help topics specified in requirements and/or design.	X=1-A/B A= Number of help topics with incomprehensive information. B= Total of help topics specified in requirements and/or design.	0<=X<=1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	
27c 28	**Completeness of Training material**	What proportion of functions is described in the training material?	Count the number of implemented functions whit training material and compare it with the total number of functions as specified in requirements.	X=1-A/B A= Number of functions whit training material. B= Total of functions whit training material as specified in requirements.	0<=X<=1 The closer to 1, the more complete.	Absolute	X=count/count A=count B=count	

Usability. Operability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
28	**Input validity checking**	What proportion of input items provide check for valid data	Count the number of input items, which check for valid data and compare with the number of input items, which could check for valid data	X=A/B A=Number of input items which check for valid data B=Number of input items which could check for valid data	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	
29	**User operation cancellability**	What proportion of functions can be cancelled prior to completion?	Count the number of implemented functions, which can be cancelled by the user prior to completion and compare it with the number of functions requiring the precancellation capability	X=A/B A=Number of implemented functions which can be cancelled by the user B= Number of functions requiring the precancellation capability	0 <= X <= 1 The closer to 1, the better cancellability	Absolute	X=count/count A=count B=count	
30	**User operation Undoability**	What proportion of functions can be undone?	Count the number of implemented functions which can be undone by the user after completion and compare it with the number of functions be undone by the user	X=A/B A=Number of implemented B= Number of functions.	0 <= X <= 1 The closer to 1, the better undoability	Absolute	X=count/count A=count B=count	
31	**Customizability**	What proportion of functions can be customized during operation?	Count the number of implemented functions, which can be customized by the user during operation and compare it with the number of functions requiring the customization capability	X=A/B A=Number of functions which can be customized during operation B=Number of functions requiring the customization capability	0 <= X <= 1 The closer to 1, the better customizability	Absolute	X=count/count A=count B=count	

32	**Physical accessibility**	What proportion of functions can be customized for access by users with physical handicaps	Count the number of implemented functions, which can be customized and compare it with the number of functions	X=A/B A=Number of functions which can be customized B=Number of functions	0 <= X <= 1 The closer to 1,the better physical accessibility	Absolute	X=count/count A=count B=count	

- Examples of physical accessibility are inability to use a mouse and blindness.

33	**Operation status monitoring capability**	What proportion of functions has operations status monitoring capability?	Count the number of implemented functions, which status can be monitored and compare it with the number of functions requiring the monitoring capability.	X=A/B A=Number of functions having status monitoring capability B=Number of functions that are required to have monitoring capability.	0 <= X <= 1 The closer to 1, the better monitoring capability	Absolute	X=count/count A=count B=count	
34	**Operational consistency**	What proportion of operations behaves the same way to similar operations in other parts of the system?	Count the number of instances of operations with inconsistent behavior and compare it with the total number of operations	X=1-A/B A=Number of instances of operations with inconsistent behavior B=Total number of r operations	0 <= X <= 1 The closer to 1, the more consistent.	Absolute	X=count/count A=count B=count	
35	**Message Clarity**	What proportion of messages is self-explanatory?	Count the numbers of implemented messages with clear explanations and compare it with the total number of messages implemented.	X=A/B A=Number of implemented messages with clear explanations. . B=Number of messages implemented	0 <= X <= 1 The closer to 1, the more clear	Absolute	X=count/count A=count B=count	

- Clear error messages explain to the user what action to take to recover from the error.

36	**Interface element clarity**	What proportion of interface elements is self-explanatory?	Count the number of interface elements which are self explanatory and compare it with the total number of interface elements	X=A/B A=Number of interface elements which are self-explanatory. B=Total number of interface elements	0 <= X <= 1 The closer to 1, the more clear.	Absolute	X=count/count A=count B=count	

- Elements are self-explanatory when they use plain text or provide “hover-help” or “tool tips”.

37	**Operational error recoverability**	What proportion of functions can tolerate user error?	Count the number of functions implemented with user error tolerance and compare it to the total number of functions requiring the tolerance capability	X=A/B A=Number of functions implemented with user error tolerance B=Total number of functions requiring the tolerance capability	0 <= X <= 1 The closer to 1, the more recoverable.	Absolute	X=count/count A=count B=count	
37a **29**	**Default value availability**	What proportion of functions and/or input items have default value availability?	Count the number of implemented functions and/or input items whit default values availability as specified in requirements and/or design.	X=A/B A= Number of functions and/or input items with default values availability. B= Total of functions and/or input items with default values availability as specified in requirements and/or design.	0<=X<=1 The closer to 1, the better default value availability.	Absolute	X=count/count A=count B=count	
37b **30**	**Operation procedure reduction availability**	What proportion of functions will permit the user to reduce operation procedures?	Count the number of implemented functions whit operation procedure reduction availability as specified in requirements and/or design.	X=A/B A= Number of functions whit operation procedure reduction availability. B= Total of functions whit operation procedure reduction availability as specified in requirements and/or design.	0<=X<=1 The closer to 1, the better procedure reduction availability.	Absolute	X=count/count A=count B=count	

Usability. Attractiveness measures.

Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure

38	**Attractive interaction**	How attractive is the interface to the user?	Questionnaire to users Questionnaire to assess the attractiveness of the interface to users.	Assessment classification	Ordinal X= Count (Count is a score)	Absolute	X=count/count A=count B=count

- This could be based on screen sketches or mock-ups.
- Issues that potentially contribute to attractiveness include: Alignment of items (vertical and Horizontal), Grouping, color and graphical design, Use of colors, Appropriate and reasonable sized graphics, Use of hit space/separators/borders, Animation, Typography, and 3D interface.

39	**User Interface appearance customizability**	What proportion of user interface elements can be customized in appearance?	Inspection (by expert)	X=A/B A=Number of types of interface elements that can be customized. B=Total number of types of interface elements.	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count

Usability. Compliance measures.

40	**Usability Compliance**	How compliant is the product to applicable regulations, standards and conventions for usability?	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.	X=A/B A= Number of correctly implemented items related to usability compliance confirmed in evaluation B= Total number of compliance items	0 <= X <= 1 The closer to 1, the more compliant.	Absolute	X=count/count A=count B=count
40a **31**	**Consistency with other systems**	How consistent is the product in relation to others known systems?	Count the number of items requiring with other knows systems and compare with the number of items requiring consistency with others know systems as in the specifications and/or design.	X=A/B A= Number correctly implemented items related to usability consistency confirmed in evaluation. B= Total number of consistency items.	0<=X<=1 The closer to 1, the more consistent.	Absolute	X=count/count A=count B=count

Efficiency measures.

Efficiency. Time behavior measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
41 **32**	**Specified task Response time**	What is the estimated time to complete a specified task?	Define the limits of the specified task. Calculate the task complexity, estimate the elements involve in task description as defined in requirements and design. Estimate the response time in function of the size of data (n).	X= Estimated software platform response time (n) + Estimated data management response time (n) + Estimated transmission response time (n) + algorithms set complexity of the specified task (n).	The shorter the better.	Ratio	X=time	Complexity. OO Complexity.

- Task: an elementary process (input, input transformation and input transformed output), with a well-defined purpose.
- It is possible to estimate task time based in task algorithm complexity analysis.
- It is a MIS systems result interpretation.

.

41a **33**	**Estimated software platform response time**	What is the estimated response time of software platform needed to complete a specified task?	Use the software platform performance data (bytes/time) and the result of measure "Estimated I/O Utilization size".	X= Sum (response time of software platform's element) * "Estimated I/O Utilization size" of specified task.	The shorter the better.	Ratio	X=time	Complexity. OO Complexity.

- Software platform = operating system, application server, transaction monitor, call center system, etc.
- Software platform performance data = response time described by the provider or technical sheets for software platform.
- It is a MIS systems result interpretation.

.

41b **34**	**Estimated data management response time**	What is the estimated response time of data management needed to complete a specified task?	Use the data management performance data (bytes/time) and the result of measure "Estimated I/O Utilization size".	X= Sum (response time of data management's element) * "Estimated I/O Utilization size" of specified task.	The shorter the better.	Ratio	X=time	Complexity. OO Complexity.

- Data management = data base management systems, file manager systems, searching systems, expert systems, etc.
- Data management performance data = response time described by the provider or technical sheets for data management systems.
- It is a MIS systems result interpretation.

.

41c 35	**Estimated transmission response time**	What is the estimated response time of transmission needed to complete a specified task?	Use the transmission performance data (bytes/time) and the result of measure "Estimated I/O Utilization size".	X= Sum (response time of transmission's element) * "Estimated I/O Utilization size" of specified task.	The shorter the better.	Ratio	X=time	Complexity. OO Complexity.

- Transmission = intranet, internet, local networks, unwired communications, etc.
- Transmission performance data = response time described by the provider or technical sheets for transmission systems.
- It is a MIS systems result interpretation.

42 36	**Throughput time**	What is the estimated number of same-type tasks that can be performed over a unit of time?	Classify tasks according their functionality. Use support software performance data, data treatment software performance data and communication software and hardware performance data and task complexity data to estimate the response time of a specific task type. Estimate the response time using one unit of size.	Y= Fixed response time of all software platform's elements (1 unit of data size) + fixed response time of all data management's elements (1 unit of data size) + fixed response time of all transmission's elements (1 unit of data size) + the algorithms set of a specified task type complexity (1 unit of data size) X = unit time / Y	The greater the better	Ratio	X=count	Complexity. OO Complexity.

- Task: an elementary process (input, input transformation and input transformed output), with a well-defined purpose.
- Software platform performance data = response time described by the provider or technical sheets for operating system, application server, transaction monitor, call center system, etc.
- Data management software performance data = response time described by the provider or technical sheets for data base management systems, file manager systems, searching systems, expert systems, etc.
- Transmission software and hardware performance data = response time described by the provider or technical sheets for intranet, internet, local networks, unwired communications, etc
- It is possible to estimate task time based in task algorithm complexity analysis.
- It is a MIS systems result interpretation.

37

Efficiency. Resource Utilization measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
44 38	**Estimated I/O Utilization**	What is the estimated I/O utilization to complete a specified task?	Count total number of task input and output events as defined in requirements, design and code.	X= Total number of individual readings events defined of coded for a specified task (inputs) + Total number of individual writings events defined of coded for a specified task (outputs).	The shorter the better.	Ratio	X=count	

- Task: an elementary process (input, input transformation and input transformed output), with a well-defined purpose.

44a 39	**Estimated I/O Utilization size**	What is the estimated I/O utilization size (buffering) in a specified task?	Count the maxim size of task data input and data output (according data type definition) as defined in requirements, design and code.	X= Total input data length of specified task (input buffer size) + Total output length data size of specified task (output buffer size).	The shorter the better.	Ratio	X=count	
45 40	**I/O Utilization Message Density**	What is the density of messages relating to I/O utilization in the lines of code responsible in making system calls?	Count the number of errors pertaining to I/O failure and warnings and compare it to number of lines of code responsible in system calls.	X=A/B A=number of I/O related error messages. B=number of lines of code directly related to system calls.	The greater the better.	Absolute	X=count/count A=count B=count	
46	**Memory utilization**	What is the estimated	Estimate the memory requirement.	X=size in bytes (calculated or simulated)	The lesser the	Ratio	X=size	

		memory size that the product will occupy to complete a specified task?			better.			
47 41	**Memory utilization message density**	What is the density of messages relating to memory utilization in the lines of code responsible in making system calls?	Count the number of error messages pertaining to memory failure and warnings and compare it to the number of lines of code responsible in system calls.	X=A/B A=Number of memory related error messages. B=Number of lines of code directly related to system calls.	The greater the better.	Ratio	X=count/count A=count B=count	

42

Efficiency. Compliance measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
49	Efficiency Compliance	How compliant is the efficiency of the product to applicable regulations, standards and conventions.	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.	X=A/B A= Number of correctly implemented items related to efficiency compliance confirmed in evaluation B= Total number of compliance items	0 <= X <= 1 The closer to 1, the more compliant.	Absolute	X=count/count A=count B=count	

Maintainability measures.

Maintainability. Analyzability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
50	**Activity recording**	How thorough is the recording of the system status?	Count the number of items logged in the activity log as specified and compare it to the number of items required to be logged.	X=A/B A=Number of implemented data login items as specified confirmed in review B=Number of data items to be logged defined in the specifications	0 <= X <= 1 The closer to 1, more data provided to record system status.	Absolute	X=count/count A=count B=count	

- It is necessary to convert this value to the <0,1> interval if making summarization of characteristic.

50a 43	**Activity recording legibility**	How useful and complete is the information recorded in the activity log?	Count the number of duplicated, unnecessary or incomplete items logged in the activity log as specified and compare it to the number of items that are required to be logged.	X=1-A/B A=Number of duplicated, unnecessary or incomplete implemented data login items confirmed in evaluation. B=Number of data items to be logged defined in the specifications.	0 <= X <= 1 The closer to 1, more legibility.	Absolute	X=count/count A=count B=count	
51	**Readiness of diagnostic function**	How thorough is the provision of the diagnostic functions.	Count the number of implemented diagnostic functions as specified and compare it to the number of diagnostic functions required in specifications.	X=A/B A=Number of implemented diagnostic functions as specified confirmed in review B=Number of diagnostic functions required	0 <= X The closer to 1, the better implementation of diagnostic functions.	Absolute	X=count/count A=count B=count	

- This metric is also used to measure failure analysis capability and causal analysis capability.
- It is necessary to convert this value to the <0,1> interval if making summarization of characteristics.

Maintainability. Changeability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
52	**Change**	Are changes to	Record ratio of module change	X=A/B	0 <= X <= 1	Absolute	X=count/count	

	recordability	specifications and program modules recorded adequately in the code with comment lines?	information control or little changes, high stability	A=Number of changes in functions/modules having change comments confirmed in review B=Total number of functions/modules changed from original code	The closer to 1, the more recordable. The change control 0 indicates poor change		A=count B=count	
52a **44**	**Modification complexity**	Which is the complexity in function modification?	Calculate the functions complexity according to a predefined scale.	X= Complexity (function) Use the technical quality measures to calculate the function complexity.	The lesser the better.	Ordinal	X= Score	Complexity sub-characteristic. OO Complexity sub-characteristic.
52b **45**	**Parameterized availability**	Which is the reusability level of implemented functions?	Count the number of functions implemented with parameterized availability and compare with the total number of functions specified.	X= A/B A=Number of functions with parameterized availability. B=Total of number of functions specified.	0 <= X <= 1 The closer to 1, the more availability.	Absolute	X=count/count A=count B=count	
52c **46**	**Reuse utilization**	What proportion of functions could be reutilized?	Calculate reusability level of implemented functions according to a predefined scale.	X= Reusability (function) Use the technical quality measures to calculate the function reusability.	The greater the better.	Ordinal	X= Score	Selfdescriptiveness, OO Maintainability, OOReusability.
52d **47**	**Programming Style consistency**	How many functions are implemented using the same programming style or programming standards?	Count the number of functions implemented using the same programming style or programming standards and compare it with the total of functions specified in requirements and/or design.	X= A/B A=Number of functions implemented using the same programming style or programming standards. B=Total of number of functions specified.	0 <= X <= 1 The closer to 1, the more consistency.	Absolute	X=count/count A=count B=count	
52e **48**	**Frameworks utilization**	What proportion of functions specified is part of a framework?	Count the number of functions that are founded in a know framework and compare it with the total of functions specified in requirements and/or design.	X= A/B A=Number of functions in a framework as view in evaluation. B=Total of number of functions specified.	0 <= X <= 1 The closer to 1, the more used. The greater the better.	Absolute	X=count/count A=count B=count	
52f **49**	**Patterns utilization**	What proportion of functions specified are part of a patterns?	Count the number of functions that are founded in a know patterns and compare it with the total of functions specified in requirements and/or design.	X= A/B A=Number of functions in a patterns as view in evaluation. B=Total of number of functions specified.	0 <= X <= 1 The closer to 1, the more used. The greater the better.	Absolute	X=count/count A=count B=count	
52g **50**	**Programs libraries utilization**	What proportion of functions use a know programs libraries?	Count the number of functions that use a know programs libraries and compare it with the total of functions specified in requirements and/or design.	X= A/B A=Number of functions that use a know programs libraries as view in evaluation. B=Total of number of functions specified.	0 <= X <= 1 The closer to 1, the more used. The greater the better.	Absolute	X=count/count A=count B=count	
52h **51**	**Data stores / procedures utilization**	What proportion of functions use an existed data stores or data procedures?	Count the number of functions that use an existed data stores or data procedures and compare it with the total of functions specified in requirements and/or design.	X= A/B A=Number of functions that use an existed data stores or data procedures as view in evaluation. B=Total of number of functions specified.	0 <= X <= 1 The closer to 1, the more used. The greater the better.	Absolute	X=count/count A=count B=count	

Maintainability. Suitability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
53	**Change impact**	What is the frequency of adverse impacts after modification?	Count the number of detected adverse impacts after modification and compare it to the number of modifications performed.	X=1-A/B A=Number of detected adverse impacts after modifications B=Number of modifications made	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	

54	**Modification impact localization**	How large is the impact of the modification on the software product?	Count the number of affected variables from a modification and compare it to the total number of variables in the product.	X=A/B A=Number of affected variable data by modification, confirmed in review B=Total number of variables	0 <= X <= 1 The closer to 0, the lesser impact of modification.	Absolute	X=count/count A=count B=count

- Impacted variable is a) all variables in the instruction that was changed b) Variable that is in the same instruction with the variable defined by (a).

54a **52**	**Technical Documentation consistency**	How many technical documents are structured in the same way to similar technical documents in the system?	Count the number of technical documents with inconsistency structure and compare it with the total of technical documents specified in requirements.	X=1-A/B A= Number of technical documents with inconsistency structure. B= Total of technical documents specified in requirements.	0<=X<=1 The closer to 1, the more consistent.	Absolute	X=count/count A=count B=count
54b **53**	**Technical Documentation understandability**	What proportion of help topics will the software developer team be able to understand correctly?	Count the number of technical documentation with incomprehensive information (in a peer review for example) and compare it with the total of technical documentation specified in requirements and/or design	X=1-A/B A= Number of technical documents with understandability problems. B= Total of technical documents specified in requirements.	0<=X<=1 The closer to 1, the more understandable.	Absolute	X=count/count A=count B=count
54c **54**	**Completeness of Technical Documentation**	What proportion of functions is described in the technical documentation?	Count the number of implemented functions whit technical documentation and compare with the total number of functions as specified in requirements.	X=1-A/B A= Number of uncompleted technical documents. B= Total of technical documents specified in requirements.	0<=X<=1 The closer to 1, the more uncompleted.	Absolute	X=count/count A=count B=count

Maintainability. Testability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
55	**Completeness of built-in test function**	How complete is the built-in test capability.	Count the number of implemented built-in test functions as specified and compare it to the number of built-in test functions in the requirements.	X=A/B A=Number of implemented built-in test function as specified confirmed in review B=Number of built-in test function required	0 <= X <= 1 The closer to 1, the more complete.	Absolute	X=count/count A=count B=count	
56	**Autonomy of testability**	How independently can the software be tested?	Count the number of dependencies on other systems for testing, that have been simulated with stubs and compare it with the total number of test dependencies on other systems.	X=A/B A=Number of dependencies on other systems for testing that have been simulated with stubs B= Total number of test dependencies on other systems	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	
57	**Test progress observability**	How complete are the built in test result displays during testing?	Count the number of implemented checkpoints as specified and compare it to the number specified checkpoints required by design.	X=A/B A=Number of implemented checkpoints as specified confirmed in review B=Number of designed checkpoints	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	

Maintainability. Compliance measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
58	**Maintainability Compliance**	How compliant is the maintainability of the product to applicable regulations, standards	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the	X=A/B A= Number of correctly implemented items related to maintainability compliance confirmed in evaluation	0 <= X <= 1 The closer to 1, the more compliant.	Absolute	X=count/count A=count B=count	

		and conventions.	specification.	B= Total number of compliance items				

Portability measures.

Portability. Adaptability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
59	**Adaptability of data structures**	How adaptable is the product to the data structure changes.	Count the number of data structures, which are operable and have no limitation after adaptation and compare it to the total number of data structures requiring adaptation capability.	X=A/B A=Number of data structures which are operable and has no limitation after adaptation, confirmed in review B=Total number of data structures requiring adaptation capability	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	
60	**Organizational Environment adaptability**	How adaptable is the product to organizational change?	Count the number of implemented functions that are capable of achieving required results in specified multiple organizational and business environments as specified and compare it to the number of functions with organizational environment adaptation capability requirements.	X=A/B A=number of implemented functions which are capable of achieving required results in specified multiple organizational and business environment as specified, confirmed in review B=Total number of functions with organizational environment adaptation capability requirements	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	
60a **55**	**Internationalization**	How adaptable is the product to internationalization aspects?	Count the number of implemented functions that are capable of achieving required results in specified multiple countries as specified and compare it to the number of functions with internationalization capability requirements.	X= A/B A=Number of functions that are capable of achieving internationalization as view in evaluation. B=Total of number of functions specified.	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	
61	**Hardware Environmental Adaptability (H/W, network)**	How adaptable is the product to the H/W related environmental change.	Count the number of implemented functions that are capable of achieving required results in specified multiple H/W environments as specified and compare it to the number of functions with H/W environment adaptation capability requirements.	X=A/B A=Number of implemented functions which are capable of achieving required results in specified multiple H/W environment as specified, confirmed in review B=Total number of functions with H/W environment adaptation capability requirements	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	
62	**System software Environmental adaptability (OS, concurrent application)**	How adaptable is the product to system software related environmental changes	Count the number of implemented functions which are capable of achieving required results in specified multiple system software environments as specified and compare it to the number of functions with system software environment adaptation capability requirements specified, confirmed in review.	X=A/B A=Number of implemented functions which are capable of achieving required results in specified multiple system software environment as B=Total number of functions with system software environment adaptation capability requirements B=Total number of functions with system software environment adaptation capability requirements	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	
63	**Porting User Friendliness**	How effortless is it to perform porting operations on the product	Count the number of implemented functions that are capable of supporting ease-of-adaptation by user as specified and compare it to the number of functions with easy-to-adapt capability requirements.	X=A/B A=Number of functions supporting ease-of-adaptation by user as specified, confirmed in review B=Total number of functions with ease-to-adapt capability requirements	0 <= X <= 1 The closer to 1, the more friendly.	Absolute	X=count/count A=count B=count	

Portability. Installability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
64	**Ease of setup retry**	How easy is it to repeat setup operation?	Count the number of implemented setup retry operations and compare it to the number of setup retry operations required	X=A/B A=Number of implemented retry operations for setup, confirmed in review B=Total number of setup operations required	0 <= X <= 1 The closer to 1, the easier.	Absolute	X=count/count A=count B=count	
64a 56	**Ease of installation**	How easy is it to install the product?	Count the number of complex implemented setup operations and compare it to the number of setup operation specified.	X= A/B A=Number of functions that have complex implemented setup operations as view in evaluation. B=Total of number of setup operations specified.	0 <= X <= 1 The closer to 1, the easier.	Absolute	X=count/count A=count B=count	
65	**Installation effort**	What level of effort is required for installation?	Count the number of implemented installation automated steps and compare it to the number of prescribed installation steps.	X=A/B A=Number of automated installation steps confirmed in review B=Number of installation steps required	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	

- Prescribed: e.g., number of windows/commands/manual operation to reach target operation).

66	**Installation flexibility**	How flexible and customizable is the installation capability?	Count the number of implemented customizable installation operations as specified and compare it to the number of installation operations with customization capability requirements	X=A/B A=Number of implemented customizable installation operation as specified confirmed in review B=Number of customizable installation operation required	0 <= X <= 1 The closer to 1, the more flexible.	Absolute	X=count/count A=count B=count	

- Customizable: e.g., nesting depth, number of panels.

Portability. Replaceability measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
67	**Continued use of Data**	What is the amount of original data that remain unchanged after replacement with this product?	Count the number of data items that continue to be used after replacement as specified, and compare it to the total number of data items required to be used from the old data after software replacement.	X=A/B A=Number of software data items that continue to be used as specified after replacement, confirmed in evaluation. B=Number of old data items required to be used from old software	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	
68	**Functional inclusiveness**	What's the amount of functions that remain unchanged?	Count the number of functions covered by new software that produces similar results and compare it to the number of function in the old software. .	X=A/B A=Number of functions covered by new software that produces similar results, confirmed in review B=Number of functions in old software	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	

Portability. Co-existence measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
69	**Available co-existence**	How flexible is the product in sharing its environment with other products without adverse impacts on	Count the number of entities with which product can co-exist as specified and compare it to the number of entities in production environment that	X=A/B A= Number of entities with which product can co-exist as specified B= Number of entities in production environment that require co-existence	0 <= X <= 1 The closer to 1, the better.	Absolute	X=count/count A=count B=count	

other products.

Portability. Compliance measures.

	Measure Name	Purpose of the measure	Method of application	Measurement, formula and data element computations	Interpretation of measured	Measure scale	Measure type	Technical measure
70	Portability Compliance	How compliant is the portability of the product to applicable regulations, standards and conventions.	Count the number of items requiring compliance that have been met and compare with the number of items requiring compliance as in the specification.	X=A/B A= Number of correctly implemented items related to portability compliance confirmed in evaluation B= Total number of compliance items	0 <= X <= 1 The closer to 1, the more complete.	Absolute	X=count/count A=count B=count	

The interventions to measures on the other hand, appear rather as the maintenance or update effort which should be recognized as a natural process, particularly important in the domain so quickly evolving as software engineering. The recommendations presented in this paper together with details to be published in the previously signalled white paper could – if recognized by ISO - help speed up updating the standard in its new edition and once again prove the value of co-operation between standardizing organizations and academic community.

6. CONCLUSIONS

The research presented in this paper found the evaluation of the state-of-the-art, relevancy and applicability exhibited by both quality model and associated internal quality measures of ISO/IEC 9126-3 as relatively positive. The identified needs for modifications indicate rather necessity of maintenance than of a thorough re-engineering of the standard.

Following this conclusion, the enhancements of the quality model proposed by this research could make a valuable enrichment while keeping the basic structure of the model and its sustained value intact.

The proposed enhancements in the area of measurable artifacts and input to measurement could make the necessary intervention allowing for keeping the standard up to date with the evolving development technologies.

The rather massive intervention proposed in the area of detailed measures (95 interventions of different type) may create a negative impression while the real conclusion would be of the following nature: the fact that only 3 measures were proposed to be deleted (less than 3%) indicates that the core of measures stay valid even if requiring adaptive adjustments. Addition of new measures makes part of perfective maintenance required to stay abreast with the technology to which the standard is supposed to be effectively applied. Strengthening prediction capabilities towards external quality makes one of the other benefits of the above interventions.

The practical implementation of suggested changes in ISO software quality measurement and evaluation standards (if they meet ISO/IEC SC7 subcommittee interest) could become beneficial to both industrial and academic users on the international level.

7. REFERENCES

[BOE76] Boehm, B.W., Brown, J.R.; Lipow, M., "Quantitative Evaluation of Software Quality", Proceedings of Second International Conference on Software Engineering, IEEE, Oct., 1976

[ESA03] “Software Quality Modeling and Metrication”; ESA Multimedia; 2003; esamultimedia.esa.int /docs/industry/SME/2003/software_engineering/B-Software_Quality_Modeling_and_Metrication.pdf

[ISO01] ISO/IEC 9126. Software Engineering - Product Quality – Part 1: Quality Model, ISO 2001

[ISO02] ISO/IEC 9126. Software Engineering - Product Quality – Part 2: External Metrics, ISO 2003

[ISO03] ISO/IEC 9126. Software Engineering - Product Quality – Part 3: Internal Metrics, ISO 2003

[ISO04] ISO/IEC 14598. Software Product Evaluation - Part 1: General Overview, ISO 1999

[ISO05] ISO/IEC 15288. Lifecycle Management. System Lifecycle Processes. 2000.

[ISO06] ISO/IEC 12207. Standard for Information Technology. Software lifecycle processes. Industry implementation of International Standard ISO/IEC 12207 : 1995.

[JAC99] Jacobson, I., Booch, G., Rumbaugh, J. (1999). The Unified software development process (1st edition). Addison-Wesley;

[LAU02] Lauesen, S. (2002). Software Requirements Styles and Techniques. Addison-Wesley.

[MEY98] Meyer, B., (1998) The role of object-oriented metrics. *A shorter variant of this article appeared in Computer (IEEE), as part of the Component and Object Technology department, in the November 1998 issue. http//:archiv.eiffel.com*

[PRE97] Pressman, R. S. (1997). Software Engineering a practitioner’s approach (4th edition). McGraw-Hill;

[PRO01] Probert, R. L. (2001). Software Quality Engineering. Telecommunication Engineering Research Group, School of Information Technology and Engineering, Faculty of Engineering, University of Ottawa; From www.site.uottawa.ca/~bob/csi5111/notes/1.0SQ1209.ppt

[RUP00] IBM and RUP (Rational Unified Process) from: http://www.rational.com/.

[SMI97] Smith, D. (1997 Compilation). Risk Management & Metrics. Class CS 3302.

From http://www.cc.gatech.edu/classes/cs3302_97_fall/

[TOP98] ISO 12207, ISO 15504, SW-CMM v1.1, SW-CMM v2 Draft C Mapping – Top-Level Standards Map. (1998). From www.sei.cmu.edu/pub/cmm/Misc/standards-map.pdf