Transactions on Information and Communications Technologies vol 9, © 1994 WIT Press, www.witpress.com, ISSN 1743-3517

Software product evaluation metrics: a methodological approach

A. Jeanrenaud & P. Romanazzi
Software Quality Laboratory Tecnopolis CSATA Novus Ortus, 70010 Valenzano (Bari), Italy

ABSTRACT

This paper reports essentials of a software product evaluation methodology, called CDSEM (Checklist Driven Software Evaluation Methodology), designed by Software Quality Laboratory of Tecnopolis CSATA Novus Ortus. Our intention is to focalize the use of software evaluation metrics in the framework of our methodology.
We consider software product as composed by different parts: software system, product documentation, user documentation, support services and distribution media. Each component needs a specific set of metrics and tools for the evaluation process. After each component has been evaluated, the methodology provides an unified assessment process.

The methodology proposes the evaluation models in accordance with the standard ISO 9126 (Information technology - Software product evaluation - Quality characteristics and guidelines for their use) taking also into account the emerging new parts of the standard. The six characteristics defined in ISO 9126 (functionality, reliability, usability, maintainability, portability, efficiency), are exploded, for every component, into sublayers of abstractions till to the identification of the measurable items (metrics). Moreover, the methodology identifies, for each metric, tools and procedures for the evaluation (code measures, inspection etc.).
A tool has been developed on PC platform (CDSET, Checklist Driven Software Evaluation Tool) to manage the methodology information base, results and reports.

INTRODUCTION

Before introducing the use of software evaluation metrics in the framework of our methodology, a brief description of the methodology is necessary in order to better explain the context in which CDSEM metrics work.

CDSEM [1] is the Software Product Evaluation Methodology implemented by Laboratorio Qualita' Software of Tecnopolis CSATA Novus Ortus in the perspective to offer services on software product and process evaluation.

ISO/IEC 9126 [3] framework is the starting point in the methodology evaluation process definition. The stages defined in the standard, quality requirement definition, evaluation preparation and evaluation procedure, are the core of the methodology.

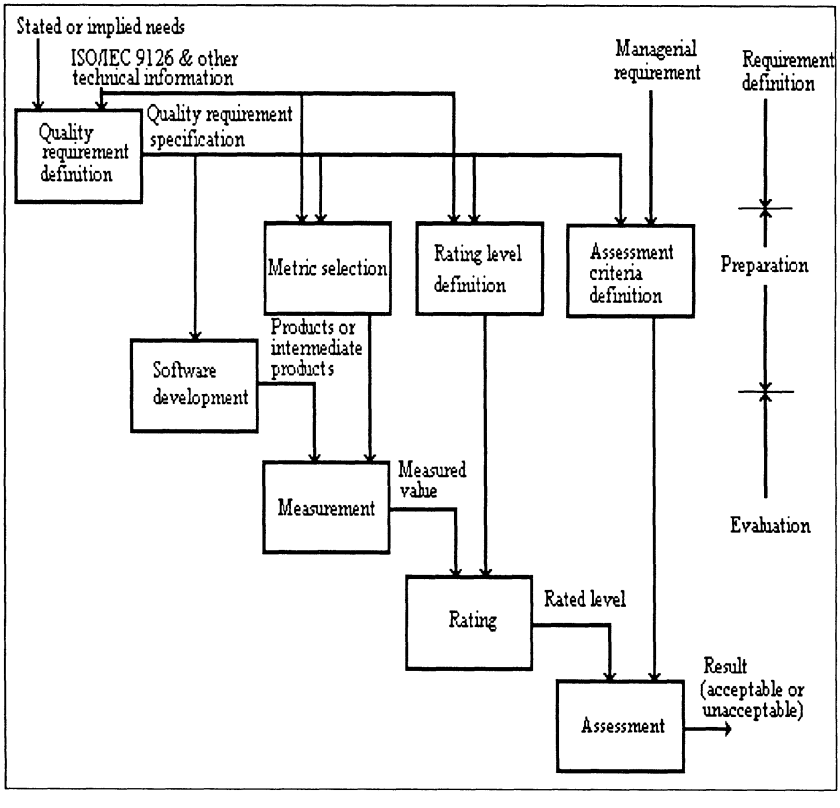

fig. 1 - ISO/IEC 9126 Evaluation process model [3]

ISO/IEC 9126 general framework is exploded in the CDSEM Evaluation Process Model:

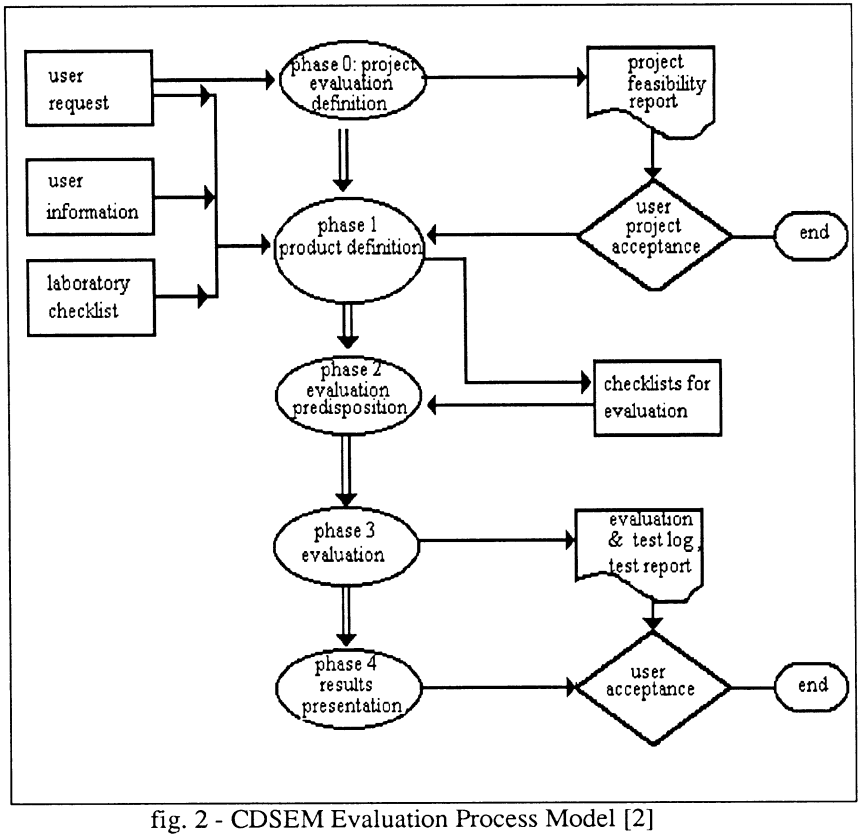

fig. 2 - CDSEM Evaluation Process Model [2]

Phase 1 of the methodology consists of the evaluation product definition. CDSEM considers software product as composed by five components:

- Software System
- Product documentation
- User documentation
- Support Services
- Distribution media

These components may be some or all present in a software product, and every one of them may be more or less essential for the user.
Software product quality value is determined from the quality values of different components and from the "weight" that the component assumes in that specific product.

$$\text{software product quality value} = \Sigma\, VC_i * PC_i \,/\, \Sigma\, PC_i$$

VC_i is the measured value of the i-component

PC_i is the weight of the i-component

fig. 3 - Determination of software product quality value

In product definition activity the product components under evaluation are identified (software, documentation, services provided,..). For each component the expected quality requirements are defined. Also, it is possible to indicate, if applicable, a specific standard whose conformity the evaluation process will measure.

Phase 2 of the methodology describes the evaluation predisposition activities: quality characteristics to evaluate, rating level to apply on the measures, assessment criteria are detailed. In this phase the checklists to apply in the evaluations activities are completed. Checklists include, for each component questions on quality requirements, quality characteristics to evaluate, standard conformity required.

User information is organised in CDSEM modules, see an example in fig 4.

COMPONENT:	Characteristic	Weight
Software		4
	functionality	4
	reliability	4
	usability	3
	efficiency	2
	maintainability	not applicable
	portability	not applicable
User Documentation		2
	functionality	4
	usability	4
	efficiency	1
Services		not applicable
Distribution media		not applicable
Note : weight is a value from 1 (poor) to 4 (high)		

fig. 4 - Example of characteristics and weights for components.

The weights defined for each characteristic contribute to the determination of the component quality value:

$$\text{component quality value} = \Sigma\, Vc_i * Pc_i \,/\, \Sigma\, Pc_i$$

Vc_i is the measured value of the i-characteristic
Pc_i is the weight of the i-characteristic.

fig. 5 - Determination of software product quality value

The measured values of the characteristics are determined by the answers (positive, negative or discard) to the checklists' questions.

Questions in the checklist require the application of metrics to measure a subcharacteristic, or an atomic level of a subcharacteristic.
Results from evaluations tools and techniques are normalised in "Y" and "N" values through a comparison of measured values with "acceptance tables".

Evaluations tools and techniques are detailed in a metric plan.

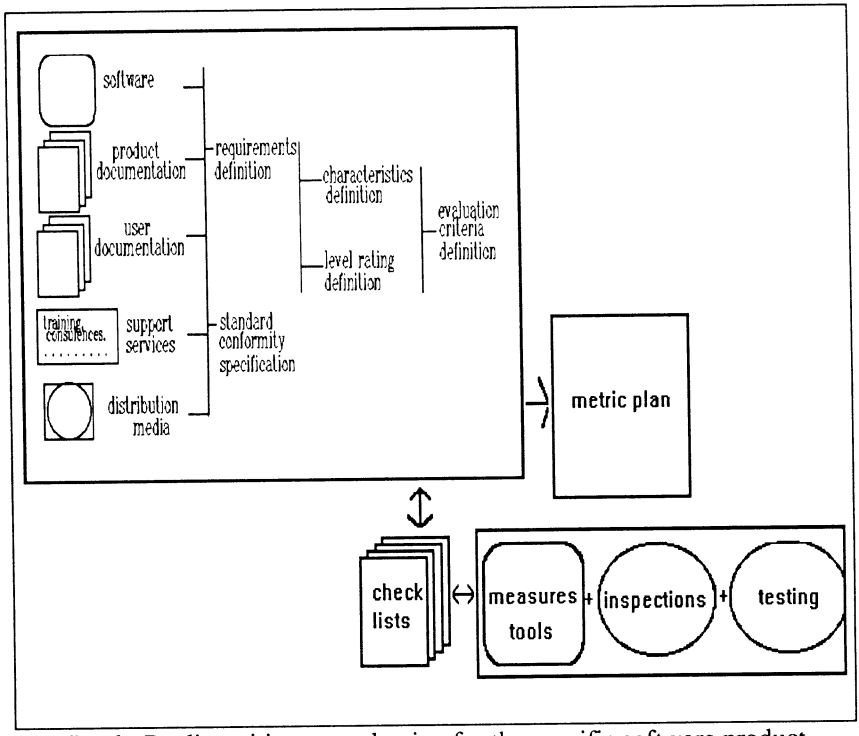

fig. 6- Predisposition to evaluation for the specific software product

Phase 3 is the evaluation phase, the software product is submitted to the evaluation activities as defined in the metric plan.

Measure's techniques are code inspections, requirements inspections, module and system test, conformity checklists, documents inspections and test, services test, distribution media test.

From the obtained measures the value for each subcharacteristc and characteristic are obtained through the following formulas:

$$Vc_j = \Sigma\, Vsc_i / nsc$$

$$|Vsc_j| = \Sigma\, m_i / (n\text{-}nd)$$

Vc_i is the measured value of the i-characteristic
Vsc_i is the measured value of the i-subcharacteristic
nsc is the number of subcharacteristics.
m_i is 1 if the i-answer is positive, otherwise is 0
n is the total number of the measures
nd is the number of discarded questions

fig. 7 - Measured values for characteristics.

QUALITY CHARACTERISTIC FRAMEWORK IN CDSEM

The peculiarity of CDSEM is to define quality characteristics for all software product components. The ISO/IEC 9126 framework is tailored utilising considerations deriving from ISO/IEC DIS 12119 Information Processing; Software Packages; Quality requirement and Testing [4] and BS 7649 - Guide to The design and preparation of documentation for users of application software [7].

The analysis of new standards on Software Product has contributed to the definition of the Quality Characteristic CDSEM framework (fig. 8).

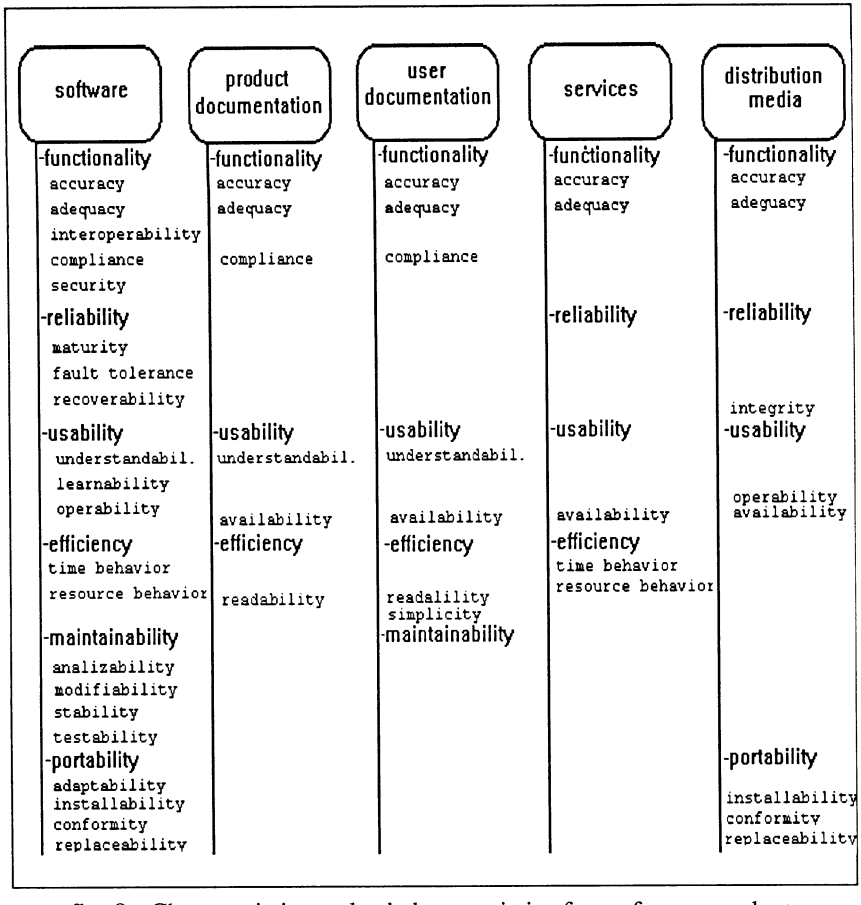

fig. 8 - Characteristics and subcharacteristics for software products

The methodology is called CDSEM because is checklists driven, the checklists are the result of the analysis of the predisposition activities, the frame for the evaluation phase, and they assure the evaluation reproducibility.

It is recommended to put highest attention in defining checklists: the more detailed they are, the more sure of valid answers you can be.

Quality characteristics selection is strictly connected with checklist questions. Questions are organised depending on the characteristic and subcharacteristic which value is involved.
Let us consider this hypothetical part of software evaluation checklist (see fig.9).
The checklist has a title, that identify the component under evaluation and the type of the checklist.

Questions are specified, for each quality characteristic and subcharacteristic to evaluate that component.

Each question has a code that identify the characteristic, the subcharacteristic and his sequence in the specific checklist

CHECKLIST PROPOSED BY THE METHODOLOGY FOR:
product component : Software
Functionality
accuracy
1.1.1.1 Has the software been realised with a minimal amount of code?
1.1.1.2 Do the function described in the user documentation produce correct results when executed with correct data?
.....
adequacy
1.1.2.1 Are executable all the functions described in the product documentation?
1.1.2.2 Are defined all referenced functions?
.....
Reliability
maturity
1.2.1.1 Is Mean Time To Failure less than "N" correct execution?
1.2.1.2 Is rate between errors found in the software and software dimension less than N/M statement?
.....

fig. 9 - Example of software quality checklist

EVALUATION METRICS SELECTION IN CDSEM

The metric plan defined in phase 2 details all the evaluation metrics selected for the software product quality evaluation. Metrics selection is executed considering the typology of the software product. Important aspects considered are those related to safety, to economy, to security, to the environment of the software product.[5].

Metrics for the characteristic "functionality" may be measured, depending on the specific software product, with different evaluation techniques, as shown in the example in fig. 10.

METRICS AND EVALUATION TECHNIQUES				
	A	B	C	D
Functionality accuracy conciseness correctness traceability	code inspect. module test trac. matrix	code inspect. module test trac. matrix	code inspect. module test inspection	-- inspection .--
adequacy	.....	.....	.....	.....
Reliability	.....	.....	.....	.....

LEGEND			
LEVELS	SAFETY ASPECTS	ECONOMY ASPECTS	SECURITY ASPECTS
A	high safety risk	financial disaster	Strategic data
B	threat to human life	large economic loss	critical data
C	damage to property	economic damage to company	protection against error risk
D	small damage to property	small economy loss	no specific risk identified

fig. 10 - Example of evaluations techniques for software product typologies

Minimum acceptable values are defined in CDSEM for each metric and for each level.

The metric plan collect the selected metrics for each characteristic, subcharacteristic of the software product components using modules whose format is in figure 11.

COMPONENT : Software CHARACTERISTIC : Functionality SUBCHARACTERISTIC: Accuracy INDICATOR : Conciseness SOFTWARE PRODUCT TYPOLOGY: Administrative [C]		
REF.CHECKLIST	1.1.1.1	
	Description	Range
METRIC	Halstead	0-1
TECHNIQUES	Static analysis	
INSTRUMENTS	Code analyser "NameTool"	
INPUT	Source code	
PROCESS	Output from "NameTool" is analysed, as defined in the Evaluation Module XXX1	

fig. 11 - Metric plan module for Conciseness

Also, measure techniques for each component are summarised in the metric plan in modules such this one:

Characteristic	subcharacter.	indicator	techniques
functionality	accuracy	conciseness	code inspection
		correctness	module test
		traceability	requirem. inspections
	adequacy	completeness	code/requir.inspection
		consistency	code/requir.inspection
	interoperab.	communicat.	system test
		data compatib.	system test
		code standardizab.	code inspection
	compliance	std complian.	conformity evaluation checklist
	security	sw access control	system test
		data access control	system test

Fig. 12 - Software functionality - Metrics for software product type C

RESULTS AND FUTURE WORK

The Methodology has been experimented on:

-Transactional Software
-High Risk Spatial Software
-OSI (open system interconnection) Software

A prototype of a support tool for the Methodology has been developed and the first release of the final product is now being implemented.

The tool guides the user through the phases of the Methodology and allows him to select quality parameters and to suite the checklists according to the product to be evaluated.

Future improvements to the methodology are planned, as the checklists refinement, the review of the acceptance tables, the assumption of new evaluations criteria regarding software products typologies.

REFERENCES

1. Jeanrenaud, A. and Romanazzi, P. "CDSEM: Checklist Driven Software Evaluation Methodology", Proceedings CQS 1993, Milano 1993

2. Jeanrenaud, A. and Romanazzi, P. "Evaluation process for software products quality: a methodology", Proceedings 3rd European conference on software quality, Madrid 1992

3. ISO/IEC 9126 -Information Technology - Software product evaluation - Quality characteristics and guidelines for their use

4. ISO/IEC DIS 12119 - Information Processing; Software Packages; Quality requirement and Testing

5. ISO/IEC CD 9126-6 - Software Product Evaluation - Part 6: Evaluators' guide

6. ISO/IEC DIS 9127 - Information processing systems, User documentation and cover information for consumer software packages

7. BS 7649 - Guide to The design and preparation of documentation for users of application software

8. Vincent, J. and Waters, A. "Software Quality Assurance, Volume II - A program guide", Prentice Hall

9. Boehm, B.W., Brown, J.R and Lipow, M. "Quantitative evaluation of software quality ", TRW System and Energy Group

10. Gilb, T. "Software Metrics", Studentlitteratur, Lund 1976

11. Watts , R. "Measuring Software Quality ", NCC Publications

12. McCabe, T "A complexity measure", IEEE Transactions of Software Engineering Vol. SE-2, No. 4 (1976)