Metric Name								
	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric	Метіс	sure	Input to ISO/IEC	C Beneficerie
	metrics		data element computations	ot measured value	scale	type	measurem 12207 ent SLCP	sv.
					•			ıce
a)	How attractive is the	How attractive is the Questionnaire to users	Questionnaire to assess the attractiveness of	<b>.</b>				·
interface	interface to the user?	~ <i>.</i>	the interface to users, after experience of					
			**************************************		- 1			
Interface	What proportion of User test	of User test	X≐A/B	0 <= X <= 1		Absolut X= Count/		
appearance	interface elements	ts	A=Number of interface elements	The closer to	ಳ	Count		
customisability	customisability can be customised in	ii	customised in appearance to user's	1, the better.		A= count		
•	appearance to the	10	satisfaction			B= count		
	user's satisfaction?		B=Number of interface elements that the					
			user wishes to customise					
			NOTE: This metric is generally used as one					
		***************************************	of experienced and justified.					
User	Does user like to u.	Does user like to use Observation of usage	User's operational frequency $X = N / OT$	X>0	Ratio	N= Count	Operation 6.5	User
operational	software frequently?		N= number of turns which user use the	The more is		OT= Time	OT= Time (test) Validation	tion
frequency			specific software functions	the better.		X= Count/ report		Human
•			OT= operation time			Time		Qualifica- interface
							User tion te	tion testing designer
			NOTE				monitorin 5.4	
			1. User may be enforced to use the software	ı			g record Operation	ion
			frequently whether user likes it or not					
			2. It is recommended to ensure whether					
			user actually feels good and is in favor of					
			using the software by interviewing end					
			user.					

NOTE:

The following complementary metrics may be used. 1) Attractive interaction

Frequency of user's feeling attractive

X = A/T

Ratio of attractive presentation

Y = (A / B)

A= Number of turns which user feel attractive during operation

Make questionnaires to user who once operate the software and analyse the answers

statistically.

A= Number of people who replied favorable answer B= Number of people who answered questionnaires

3) Favorable user popularity Favorable user popularity X = A / B \* 100(%)  $0 \le X \le 100(\%)$  The closer to 100% is the better.

B= Number of turns which user is encountering presentations

T= User operating time (observation period)

 $0 \le X$  The larger is the better.  $0 \le Y \le 1$  The closer to 1.0 is the better.

Favorable input/output expression selection availability X= 1 - (A / N) 2) Favorable input/output expression selection availability

A= Number of turns which user failed to select input/output expression

N= Number of turns which user tried to select input/output expression

Input/output expression may include multiple modal expressions such are fill-in-the-box, button, list-box, text, graph, map, pictogram, color, visual picture, voice

recognition/reading and so on.

 $0 \le X \le 1$  The closer to 1.0 is the better.

Table 7.3.5 Compliance metrics (compliance for usability)

Metric Name	Purpose	Method of application	Measurement, formula and data element computations	Interpretation Metric Measure Input to ISO/IEC of measured scale type measurem 12207	Metric scale	Measure type	Input to measurem	ISO/IEC 12207	Beneficeri es
				valuc	type		ent	SLCP Reference	
Satisfaction	How completely	Previously specify	Ratio of satisfied compliance items relating 0<= X <=1	0<= X <=1	Absolut	A= Count	Absolut A= Count Specifica- 5.3	5.3	Supplier
coverage of	does the software	required compliance items to usability	to usability	The closer to e.	<i>ن</i>	B= Count	B= Count tion of	Qualifica-	1
compliance	adhere the standards, based on standards,	based on standards.		1 is the better.		X= Count/	X= Count/ complian- tion testing 11ser	tion testing	User
items relating	conventions, style	conventions, style guides	X = 1 - (A/B)			Count	ce and		5
to usability	guides or regulations	guides or regulations or regulations relating to					related	6.5	
farman or	relating to usability?	relating to usability? usability which to be	A= Number of failed compliance items				standards,	Validation	
		adhered by software.	during testing				conven-		
							tions, style		
		Design test cases in	B= Number of total compliance items				guides or		
		accordance with					regulation		
		compliance items.	NOTE:				s		
			It may be useful to collect several measured						
		Conduct functional testing	Conduct functional testing values along time, to analyse the trend of				Test		
		for these test cases.	increasing satisfied compliance items and				specifica-		
			to determine whether they are fully satisfied	_			tion and		
			or not.				report		

© ISO/IEC TR 9126

### 7.4 Efficiency metrics

An external efficiency metric should be able to measure such attribute as the behavior of computer system including software during testing or operating.

It is recommended that the maximal and distribution time are to be investigated for many cases of testing or operating, because the measure is affected strongly and fluctuated by condition of use, such are load of processing data, frequency of use, number of connecting sites and so on. Therefore, efficiency metrics may include the ratio of measured actual value with error fluctuation to the designed value with allowed error fluctuation range which are required in specification.

It is recommended to list and to investigate the role played by factors such as "CPU" and memory load by other software's, network traffic, scheduled background processes. Possible fluctuations and valid ranges for measured values should be established and compared to requirement specifications."

It is recommended that a task is identified and defined to be suitable for software application, for examples: a transaction as a task for business application; a switching or data packet sending as a task for communication application; an event control as a task for control application; an output of data produced by user callable function for common user application.

### NOTE:

- 1. Response time: Time needed to get the result from pressing a transmissionkey. This means that response time includes processing time andtransmission time. Response time is applicable only for a interactive system. There is no significant difference when it is a standalone system. However, in the case of Internet system or other real time system, sometimes transmission time is much longer.
- 2. Processing time: The elapsed time in a computer after receiving a message to sending the result. Sometimes it includes operating overhead time, other times it only means time used for an application program.
- 3. Turn around time: Time needed to get the result from request. In many cases one turn around time includes many response. For example, in a case of banking cash dispenser, turn around time is a time from pressing initial key until you get money, mean while you must select type of transaction and wait a message, input password and wait next message etc.

### 7.4.1 Time behavior metrics

An external time behavior metric should be able to measure such attribute as the time behavior of computer system including software during testing or operating.

ISO/IEC TR 9126-2 © ISO/II

### 7.4.2 Resource utilization metrics

An external resource utilization metric should be able to measure such attribute as the utilized resources behavior of computer system including software during testing or operating.

### 7.4.3 Compliance metrics

An external efficiency compliance metric should be able to measure an attribute such as the number of functions with, or occurrences of compliance problems, which is of the software product to failing to adhere to standards, conventions or regulations relating to efficiency which are required to be adhered.

Table 7.4.1 Time behavior metrics

Response time								
Metric Name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation Metric of measured scale value type	tric Measure le type e	Input to measurem ent	ISO/IEC 12207 SLCP Reference	Beneficerie s
Response time	What is the wait time the user experiences when issuing a request and for the application to finish the process? How long does it take for users to wait for next response from the software?		Start a specified task. Response time  Measure the time it takes T = ( time of gaining the result) for the sample to complete - ( time of command entry finished) its operation.  Keep a record of each attempt.  NOTE: It is recommended to take account of time bandwidth and to use statistical analysis with measures for a lot of tasks (sample shots) not for only one task.	0 < T Ratio The sooner is the better.	io T= Time	Testing report Operation report showing elapse time		5.3 User Sys./Sw. Integration Developer 5.3 Qualifica- Maintainer tion testing 5.4 Operation 5.5 Mainte- nance
Mean response fulfillment ratio		What is the average Execute a number of wait time the user concurrent tasks (or experiences once sample shots of these issuing a request and tasks).  Its completion within Measure the time it takes the specified load to complete the selected upon the system in operation(s) in the given terms of concurrent traffic.  Tasks and system Keep a record of each utilisation?  Attempt.	Mean response time satisfaction  X = Tmean / TXmean  Tmean = \( \times \) (for i=1 to N)  TXmean = required mean response time  Ti= response time for i-th evaluation (shot)  N= number of evaluations (sampled shots)  NOTE: Required mean response time can be derived from specification of required real-time processing, user expectation of business needs or observation of user reaction. A user cognitive aspect of human ergonomics may be considerable.	0 <= X The nearer to 1 and less than 1 is the better.	Absolute X= Time/	Testing report Operation report showing elapse time	1	5.3 User Sys./Sw. Integration Developer 5.3 Qualifica- Maintainer tion testing tion testing Operation 5.4 Mainte- nance

Response time									
Metric Name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation Metric of measured scale value type	Metric scale type	Measure type	Input to measurem ent	ISO/IEC 12207 SLCP	Beneficerie s
Worst case response time ratio		What is the absolute Calibrate the test.  Ilmit on time Emulate a condition required in fulfilling whereby the system reaches a maximum load situation. Run application In the worst case, can and monitor result(s) user still get response within the specified time limit?  In the worst case, can use still get reply from the software within enough short time to be tolerable for user?	X= Tmax / Rmax  Tmax= MAX(Ti) (for i=1 to N)  Rmax = required maximum response time  MAX(Ti)= maximum response time among evaluations N= number of evaluations (sampled shots) Ti= response time for i-th evaluation (shot) NOTE:  1. Distribution may be measured like below.  Statistical maximal ratio Y= Tdev / Rmax Tdev = Tmean + K (DEV) Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation.  K: coefficient (2 or 3) DEV=SQRT{∑(Ti-Tmean) **2) / (N-1)} (for i=1 to N)	0 < X The nearer to 1 and less than 1 is the better.		Absolute X= Time/ Testing Time report report showing elapse time	Testing report Operation report showing elapse time		Sys./8w. Integration Developer 5.3 Qualifica- Maintainer tion testing 5.4 Operation 5.5 Mainte- nance
			Tmean = $\Sigma(Ti)$ / N. (for i=1 to N) TXmean = required mean response time						

Carrier									
Metric Name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation Metric of measured scale	Metric scale	Measure type	Input to ISO/IE measurem 12207	ISO/IEC 12207 ST CB	Beneficerie s
				value	iype		cur	SLCF Reference	
Throughput	How many tasks can Calibrate each task	Calibrate each task	X = A/B	X > 0	Ratio	=X •	Testing	5.3	User
ume	be successiunty performed over a	according to the intended priority given.	A = number of completed tasks B = observation time period	The larger is the better.		Count/ Time	report	Sys./Sw. Integration	Developer
	given period of	Start several job tasks.					Operation	5.3	5.3
	time?	Measure the time it takes						Qualifica-	Qualifica- Maintainer
		for the measured task to					showing	tion testing	200
		complete its operation.						5.4	SQA
		Keep a record of each						Operation	
		attempt.						5.5	
								Mainte-	
								nance	
Mean	What is the average	Calibrate each task	X = Xmean / Rmean	X > 0	Absolute	Absolute X = Time Testing	Testing	5.4	User
throughput	number of concurrent	number of concurrent according to intended		The larger is		/ Time	report	Operation	
fulfillment rati	fulfillment ratio tasks the system can	priority.	$Xmean = \Sigma(Xi)/N$	the better.				5.5	Developer
	handle over a set	Execute a number of	Rmean = required mean throughput				Operation	Mainte-	
	unit of time?	concurrent tasks.					report	nance	Maintainer
		Measure the time it takes	Xi = Ai/Ti				showing		
		to complete the selected	Ai = number of concurrent tasks observed				elapse		SQA
		task in the given traffic.	over set period of time for i-th evaluation				time		,
		Keep a record of each	Ti = set period of time for i-th evaluation						
		attempt	N = number of evaluations						

Throughput									
Metric Name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation Metric of measured scale value type	Metric scale type	Measure	Input to measurem ent	ISO/IEC 12207 SLCP Reference	Beneficeric s
Worst case throughput ratio	What is the absolute Calibrate the test. limit on the system in Emulate the condititerms of the number whereby the system	What is the absolute Calibrate the test. limit on the system in Emulate the condition terms of the number whereby the system	X = Xmax / Rmax Tmax = MAX(Xi) (for i = 1 to N)	0 < X The larger is the better.	Absolute	Absolute X = Time Testing / Time report	Testing report	5.4 Operation 5.5	User Developer
	and handling of concurrent tasks as throughput?	reaches a situation of maximum load. Run job tasks concurrently and	Rmax = required maximum throughput. MAX(Xi) = number of job tasks for i-th evaluation.				Operation Maintereport nance showing	Mainte- nance	Maintainer
		monitor result(s).	Xi = Ai / Ti Ai = number of concurrent tasks observed over set period of time for i-th evaluation Ti = set period of time for i-th evaluation N= number of evaluations. NOTE.				elapse time		SQA
			1. Distribution may be measured like below.  Statistical maximal ratio Y= Xdev / Xmax						
			Xdev = Xmean + K (DEV) Xdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation.						
			K: coefficient (2 or 3) DEV=SQRT{ \(\subseteq \text{(Ni-Xmean) **2}\) / (N-1)\} (for i=1 to N) Xmean = \(\subseteq \text{(Xi)}\)N	<i>ar</i> a.					

Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric	Metric	Measure	Input to	ISO/IEC	Beneficerie
	metrics		data element computations	of measured scale	scale	type	measurem		S
				value	type		ent	SLCP Peference	
Turnaround	What is the wait time Calibrate the test	Calibrate the test	T = Time between user's finishing	0 < T	Ratio	T= Time	Testing	5.3	User
time	the user experiences accordingly.	accordingly.	nothing output recults and mean's	The shorter			renort	Sve /Sw	
)	when issuing an	Start the job task.	genning output results and used s	the better			100	oys., ow. Integration Developmen	Devolono
	instruction to start a	instruction to start a Measure the time it takes	Inishing request.				Oneration		dona
	group of related tasks for the job task to	for the job task to	NOTE: It is recommended to take account of time bandwidth and to use statistical				report		Maintaine
	and the completion	complete its operation.	or unit bandwidth and to use statistical				showing	tion testing	
	of these related	Keep a record of each	dialysis will incasilies for a fot of tasks (sample shots) not for only one task (shot)				elapse	5.4	SOA
	tasks?	attempt.	المستلكاء عبيرت المستراط المستراك المسترك المسترك المسترك المسترك المسترك المستراك المستراك المسترك المسترك المسترك المسترك المست				time	Operation	,
								5.5	
								Mainte-	
								nance	
Mean	What is the average	Calibrate the test.	X = Tmean/TXmean	X>0	Absolute	Absolute X= Time/	Testing	5.3	User
turnaround	wait time the user	Emulate a condition		The shorter is		Time	report	Sys./Sw.	
fulfillment rati	fulfillment ratio experiences when	where a load is placed on	Tmean = $\Sigma(Ti)/N$ , (for i=1 to N)	the better.				Integration Developer	Develope
	issuing an instruction	issuing an instruction the system by executing a	TXmean = required mean turnaround time				Operation	5.3	•
	to start a group of	number of concurrent	Ti = turnaround time for i-th evaluation				report	Qualifica- Maintainer	Maintain
	related tasks and the	tasks (sampled shots).	(shot)				showing	tion testing	
	completion within a	Measure the time it takes	N = number of evaluations (sampled shots)				elapse	5.4	SQA
	specified load the	to complete the selected					time	Operation	
	system has in terms	job task in the given						5.5	
	of concurrent tasks	traffic.						Mainte-	
	and system	Keep a record of each						nance	

Table 7.4.1 Time behavior metrics (Continued)

Turnaround time

Metric Name Purpose of the Method of application Measurement

of measured scale type measurem 12207 value type ent Reference  0 < X Absolute X= Time/ Testing 5.4  The nearer to Time report Operation 1 and less turnaround time than 1 is the Time report nance time caputation  1 evaluation  1 evaluation  1 mean time to r3 times of r3 times r3	Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric	Metric	Measure	Input to	ISO/IEC	Reneficerie
what is the absolute Calibrate the test.  A limit on time  Familiate a condition  required in fulfilling  where by the system  a job task?  In the worst case,  the result of the condition  required in fulfilling  where by the system  a job task?  In the worst case,  the remaining the reduced Rawinum turnaround time  for software system  wonthor result(s).  In the worst case,  the remaining to perform a performed Run the remaining the remaining to perform a precipical tasks?  In Distribution may be measured like  below.  Statistical maximal ratio Y = Tdev / Rmax  Take = In the worst case,  the particular time; c.g. 2 or 3 times of statistical maximal ratio or the particular time; c.g. 2 or 3 times of standard deviation.  K. coefficient (2 or 3)  DEV-SQRIY (2 (TI-Tmean) **2 / (N-1));  (for i=1 to N)  Thream = Lequired mean turnaround time.		metrics		data element computations	of measured	scale		measurem	12207	S
what is the absolute Calibrate the test. X= Tnax / Rmax  d limit on time Emulate a condition required in fulfilling where by the system Tmax=MAX(T) (for i=1 to N)					value	type	•	ent	SLCP	
If the nearer to Time report Operation required in fulfilling where by the system T max= MAX(TT) (for i=1 to N) 1 and less 5.5  a job task? reaches maximum load in Rmax = required maximum turnaround time transfer tasks teaches maximum to tasks in the worst case, performed Run the MAX(TI)= maximum turnaround time better. Time report nance for software system monitor result(s). No TE:  In the worst case, performed Run the mong evaluations (sampled shots) Time report nance for software system monitor result(s). No TE:  In the worst case, performed Run the mong evaluations (sampled shots) Time report nance to perform specified among evaluation (shots)  NOTE:  In Distribution may be measured like below.  Statistical maximal ratio Y = Tdev / Rmax  Telev = Tmean + K (DEV )  Telev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation.  K. coefficient (2 or 3)  DEV=SQRY{ X (Ti-Tmean) **2 / (N-1)}  Thean = Equired mean turnaround time.	Worst case	What is the absolute	1	X= Tmax / Rmax	X>0	Absolute	X= Time/	Testing	5.4	User
a job task?  Interpretation of tasks temps of tasks time terms of tasks and MAX(Ti)= maximum turnaround time for software system monitor result(s).  No performs specified anong evaluations (sampled shots)  Ti= turnaround time for i-th evaluation (sampled shots)  No perform specified asks?  No perform specified the state of evaluation tasks?  No perform specified asks?  Ti= turnaround time for i-th evaluation (sampled shots)  Ti= turnaround time for i-th or i-th evaluation (sampled shots)  Time and the particular time (sampled shots)  Time and time for i-th or i-	turnaround	imit on time required in fulfilling		Theory MANYTH (f. 12)	The nearer to		Time	report	Operation	
time  MAX(Ti)= maximum turnaround time among evaluations N= number of evaluations (sampled shots) Ti= turnaround time for i-th evaluation (shot)  NOTE:  1. Distribution may be measured like below. Statistical maximal ratio Y= Tdev / Rmax Tdev = Tmean + K (DEV) Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation. K: coefficient (2 or 3) DEV=SQRT { \( \tilde{\text{S}}\) (Ti-Tmean) **2 \( \tilde{\text{C}}\) (N-1)} (for i=1 to N) Tmean = \( \tilde{\text{S}}\) (Ti)-N, (for i=1 to N) TXmean = required mean turnaround time	ting ratio	a job task?			I and less		Tmax	Onerotion	5.5 Mointo	Developer
MAX(Ti)= maximum turnaround time  among evaluations N= number of evaluations (sampled shots) Ti= turnaround time for i-th evaluation (shot)  NOTE: 1. Distribution may be measured like below. Statistical maximal ratio Y= Tdev / Rmax Tdev = Tmean + K (DEV) Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation. K: coefficient (2 or 3) DEV=SQRT{ \( \Subseteq \Superigonia \Sigma \text{ for i=1 to N} \) Tmean = \( \Subseteq \Superigonia \Sigma \text{ time deman turnaround time} \) Txmean = \( \Superigonia \Superigonia \text{ for i=1 to N} \) Txmean = \( \Superigonia \Superigonia \text{ time} \)			terms of tasks		better.		Time	report	nance	Maintainer
among evaluations (sampled shots)  Ti= turnaround time for i-th evaluation (shot)  NOTE:  1. Distribution may be measured like below.  Statistical maximal ratio Y= Tdev / Rmax  Tdev = Tmean + K (DEV)  Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation.  K: coefficient (2 or 3)  DEV=SQRT { \( \subseteq \text{ E}(\text{ Ti-Tmean}) **2 \) / (N-1) } (for i=1 to N)  Tmean = \( \subseteq \text{Ti} \) / N. (for i=1 to N)  Txmean = required mean turnaround time		In the worst case, how long does it take	performed. Run the e selected job task and	MAX(Ti)= maximum tumaround time			Tdev =	showing		
N= number of evaluations (sampled shots)  Ti= turnaround time for i-th evaluation (shot)  NOTE:  1. Distribution may be measured like below.  Statistical maximal ratio Y= Tdev / Rmax  Tdev = Tmean + K ( DEV )  Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation.  K: coefficient (2 or 3)  DEV=SQRT{ \(\Sigma(\text{Ti})\text{Timean}\) **2) / (N-1)}  (for i=1 to N)  Tmean = \(\Sigma(\text{Ti})/N\), (for i=1 to N)  Txmean = required mean turnaround time		for software system	monitor result(s).	among evaluations			1 IIIIc	clapse		SŲA
		to perform specified		N= number of evaluations (sampled shots)						
NOTE:  1. Distribution may be measured like below.  Statistical maximal ratio Y= Tdev / Rmax  Tdev = Tmean + K (DEV)  Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation.  K: coefficient (2 or 3)  DEV=SQRT { \( \tilde{\text{C}}\) (Ti-Tmean) **2 \) (N-1)} (for i=1 to N)  Tmean = \( \tilde{\text{C}}\) (Ti-Tmean explired mean turnaround time)		lasks?		<pre>li=turnaround time for i-th evaluation (shot)</pre>						
1. Distribution may be measured like below.  Statistical maximal ratio Y= Tdev / Rmax  Tdev = Tmean + K (DEV)  Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation.  K: coefficient (2 or 3)  DEV=SQRT{ \( \Subseteq \text{Timean} \)} \( (for i=1 to N) \)  Tmean = \( \Superagraphi \) (for i=1 to N)  TXmean = required mean turnaround time				NOTE:						
Statistical maximal ratio Y= Tdev / Rmax  Tdev = Tmean + K ( DEV )  Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation.  K: coefficient (2 or 3)  DEV=SQRT{ \( \Subseteq \text{(Ti-Tmean) **2} \) / (N-1) \( \text{for i=1 to N} \)  Tmean = \( \Subseteq \text{(Ti) / N, (for i=1 to N)} \)  Txmean = required mean turnaround time				1. Distribution may be measured like						
Statistical maximal ratio Y= Tdev / Rmax  Tdev = Tmean + K ( DEV )  Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation.  K: coefficient (2 or 3)  DEV=SQRT{Σ(Ti-Tmean) **2}/(N-1)}  (for i=1 to N)  Tmean = Σ(Ti)/N, (for i=1 to N)  TXmean = required mean turnaround time				ociow.						
Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation.  K: coefficient (2 or 3)  DEV=SQRT{ ∑( Ti-Tmean) **2) / (N-1)}  (for i=1 to N)  Tmean = ∑(Ti) / N, (for i=1 to N)  TXmean = required mean turnaround time				Statistical maximal ratio Y= Tdev / Rmax						
Tdev is time deviated from mean time to the particular time: e.g. 2 or 3 times of standard deviation. K: coefficient (2 or 3) DEV=SQRT{ \(\Sigma(\text{Ti-Tmean}) **2) / (N-1)\}\) (for i=1 to N) Thean = \(\Sigma(\text{Ti}) / \text{N}, \(\text{for i=1 to N}) TXmean = \(\text{required mean turnaround time}\)				Tdev = Tmean + K (DEV)						
the particular time: e.g. 2 or 3 times of standard deviation.  K: coefficient (2 or 3)  DEV=SQRT{ $\Sigma$ ( Ti-Tmean) **2) / (N-1)}  (for i=1 to N)  Thean = $\Sigma$ (Ti) / N, (for i=1 to N)  TXmean = required mean turnaround time				Tdev is time deviated from mean time to						
standard deviation.  K: coefficient (2 or 3)  DEV=SQRT{ $\Sigma$ (Ti-Tmean) **2) / (N-1)}  (for i=1 to N)  Tmean = $\Sigma$ (Ti) / N, (for i=1 to N)  TXmean = required mean turnaround time				the particular time: e.g. 2 or 3 times of						
K: coefficient (2 or 3) $DEV=SQRT\{ \Sigma ( Ti-Tmean) **2 \} / (N-1) \}$ $(for i=1 to N)$ $Tmean = \Sigma (Ti) / N, (for i=1 to N)$ $TXmean = required mean turnaround time$				standard deviation.						
$DEV=SQRT\{ \Sigma((Ti-Tmean)**2)/(N-1)\}$ $(for i=1 to N)$ $Tmean = \Sigma(Ti)/N, (for i=1 to N)$ $TXmean = required mean turnaround time$				K: coefficient (2 or 3)						
(for i=1 to N)  Tmean = ∑(Ti) / N, (for i=1 to N)  TXmean = required mean turnaround time				DEV=SQRT{ $\Sigma$ ((Ti-Tmean) **2) / (N-1)}	~~					
Thean = $\Sigma(\text{Ti})$ / N, (for i=1 to N) TXmean = required mean turnaround time				(for i=1 to N)						
TXmean = required mean turnaround time				Tmean = $\Sigma(Ti)/N$ , (for i=1 to N)						
				TXmean = required mean turnaround time						

Table 7.4.2 Resource	2 Resource ut	Table 7.4.2 Resource utilisation metrics						
Metric Name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation Metric of measured scale value type	Metric Measure scale type type	e Input to measurem ent	ISO/IEC 12207 SLCP Reference	Beneficerie s
UO devices utilisation satisfaction	Is software system capable to perform tasks without using I/O devices long time?	Execute concurrently a lot of tasks and observe time of I/O devices occupied.	ot I/O devices utilisation X = A / B  c A = time of I/O devices occupied  B = specified time which is designed to occupy I/O devices	0 <= X <= 1 , The less than and nearer to the 1 is the better.	Absolute A= Time B= Time X= Time/ Time	ne Testing ne report ne/ Operation report	i	5.3 Developer Qualifica- tion testing Maintainer 5.4 Operation SQA Mainte- nance
Mean I/O What is the avorable fulfillment ratio number of I/O related error messages and failures over a specified lengt time and specific utilisation?	What is the average on number of 1/O related error messages and failures over a specified length of time and specified utilisation?	Calibrate the test condition. Emulate a condition whereby the system reaches a situation of maximum load. Run the application and record number of errors due to I/O failure and warnings.	X = Amean / Rmean  Amean = \( \subseteq \text{Ai} \)/N  Rmean = required mean number of \( \text{I/O} \)  messages.  Ai = number of \( \text{I/O} \) messages for ith  evaluation  N = number of evaluations	0<= X The smaller is the better	Absolute X = Count/	Testing report Operation report showing elapse time	5.3 Qualification testing 5.4 Operation 5.5 Mainte-	5.3 User Qualifica- tion testing Developer 5.4 Operation Maintainer 5.5 Mainte- SQA
User waiting time of I/O devices utilisation	Can user perform tasks without waiting for finish of I/O devices operation?	Can user perform Execute concurrently a lot tasks without waiting of tasks and observe time for finish of I/O of I/O devices occupied devices operation? with user environment.	lot User waiting time of I/O devices utilisation 0 < T T = Time spent to wait for finish of I/O The sidevices operation NOTE: It is recommended that the maximal and distributed time are to be investigated for several cases of testing or operating, because the measures are tend to be fluctuated by condition of use.	horter is xtter.	Ratio T= Time	Testing report Operation	5.3 Qualification testin 5.4 Operation 5.5 Maintenance	5.3 User Qualification testing Developer 5.4 Operation Maintainer 5.5 Mainte- SQA nance

Table 7.4.2 Resource utilisation metrics (continued)

I/O devices resource utilization	rrce utilization							
Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric Measure Input to ISO/IEC	Measure	Input to	ISO/IEC	Beneficerie
	metrics		data element computations	of measured scale	type	measurem 12207	12207	s
				value type		ent	SLCP	
							Reference	
Visible I/O	How many I/O errors Calibrate the test	Calibrate the test	X = A/T	0 <= X Ratio		Testing	5.3	User
utilisation	were experienced	conditions. Emulate a	A = number of warning messages or system The smaller is	The smaller is	T = Time report	report	Qualifica-	
	over a set period of	condition whereby the	failures	the better	11 ×		tion testing Maintainer	Maintaincr
	time and specified	system reaches a situation	T = User operating time during user		Count/	Operation	5.4	
	resource utilisation?	of maximum I/O load.	observation		Time	report	Operation SQA	SQA
		Run the application and				showing	5.5	
		record number of errors				clapse	Mainte-	
		due to I/O failure and				time	nance	
		warnings.						
Worst case I/O	Worst case I/O What is the absolute Calibrate the test	Calibrate the test	X = Amax / Rmax	0<= X Absolu	Absolute $X =$	Testing	5.3	User
utilisation	limit on I/O required	limit on I/O required condition. Emulate a		The smaller is	Count/	report	Qualifica-	
	in fulfilling a	condition whereby the	Amax = MAX(Ai), (for $i = 1$ to N)	the better	Count		tion testing Developer	Developer
	function?	system reaches a situation				Operation		
		of maximum load. Run	MAX(Ai) = Maximum number of I/O			report	Operation	Operation Maintainer
		application and monitor	messages from 1st to i-th evaluation.			showing	5.5	
		result(s).	N= number of evaluations.			elapse	Mainte-	SQA
						time	nance	

Table 7.4.2 Resource utilisation metrics (continued)

Memory resource utilization									
Metric Name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Metric scale type	Measure type	Input to measurem ent	ISO/IEC 12207 SLCP Reference	Beneficerie s
Mean memory fulfillment ratio	Mean memory What is the average fulfillment ratio number of memory related error messages and failures over a specified length of time and a specified load on the system?	Calibrate the test condition. Emulate a condition whereby the system reaches a situation of maximum load. Run the application and record number of errors due to memory failure and warnings.	X = Amean / Rmean  Amean = \(\Sigma(\text{i})/\text{N}\)  Rmean = required mean number of memory messages.  Ai = number of memory messages for i-th evaluation  N = number of evaluations	0<= X The smaller is the better	Absolute X = Cou	nt/ nt	Testing report Operation report showing elapse time	5.3 Qualification testing 5.4 Operation 5.5 Mainten	5.3 User Qualifica- tion testing Developer 5.4 Operation Maintainer 5.5 Mainte- SQA nance
Visible memory utilisation	Visible memory How many memory utilisation crrors were experienced over a set period of time and specified resource utilisation?	Calibrate the test conditions. Emulate a condition whereby the system reaches a situation of maximum load. Run the application and record number of errors due to memory failure and warnings	X = A / T  A = number of warning messages or system. The smaller is failures  T = User operating time during user observation	0 <= X The smaller is the better	Ratio	A = Count Testing T = Time report X = Count Operati Time report showing elapse time	Testing report Operation report showing elapse time	5.3 User Qualifica- tion testing Main 5.4 Operation SQA 5.5 Mainte- nance	5.3 User Qualifica- tion testing Maintainer 5.4 Operation SQA 5.5 Mainte- nance
Worst case memory utilisation	What is the absolute limit on memory required in fulfilling a function?	Calibrate the test condition. Emulate a condition whereby the system reaches a situation of maximum load. Run application and monitor result(s)	X = Amax / Rmax  Amax = MAX(Ai). (for i = 1 to N)  Rmax = required maximum memory  messages  MAX(Ai) = Maximum number of memory  messages from 1st to i-th evaluation.  N= number of evaluations.	0<= X The smaller is the better	Absolute	X = Count/	Testing report Operation report showing elapse time	į.	5.3 User Qualifica- tion testing Developer 5.4 Operation Maintainer 5.5 Mainte- SQA nance

ISO/IEC TR 9126-2

Metric Name Pur me me Mean Wh transmission nul								***
Mean Wi transmission nu	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation Metri of measured scale value type	Metric Measure scale type type	Input to measurem ent	ISO/IEC 12207 SLCP Reference	Beneficeric s
	What is the average number of		X = Amean / Rmean	lcr is	Absolute X = Count/	Testing report	5.3 User Qualifica-	User
fulfillment ratio <sup>U2</sup> err fai	fulfillment ratio transmission related error messages and failures over a specified length of	condition whereby the system reaches a situation of maximum load. Run the application and record	Amean = $\sum (A_1)/N$ Rmean = required mean number of transmission related error messages and failures	the better	Count	Operation report showing	tion testing 5.4 Operation 5.5	Developer Maintainer
tin uti	time and specified utilisation?	number of errors due to Transmission failure and warnings.	Ai = Number of transmission related error messages and failures for i-th evaluation N = number of evaluations			elapse	Mainte- nance	SQA
Transmission Is	Is software system capable to perform	Execute concurrently specified tasks with	Transmission capacity utilisation satisfaction $X = A / B$		Absolut A= Size	Testing report	5.3 Qualifica-	Developer
utilisation tas	tasks within expected transmission	tasks within expected multiple users, observe transmission	A = transmission capacity	The less than and nearer to	X= Size / Size	e/ Operation		tion testing Maintainer 5.4
	capacity?	compare specified one.	B = specified transmission capacity which is designed to be used by the software	the 1 is the better.		report	Operation 5.5	SQA
			during execution.  NOTE: It is recommended to measure dynamically peaked value with multiple				Mainte- nance	
	;		users.					
Visible	How many	Calibrate the test	X=A/T		Ratio A = Cc	A = Count Testing	5.3	Oser
transmission tra	transmission error messages were	conditions. Emulate a condition whereby the	A = number of warning messages or system The smaller is failures	The smaller is the better	n. = X = X	ne report	Qualifica- tion testing	Qualifica- tion testing Maintainer
	experienced over a set period of time		T = User operating time during user observation		Count/ Time		5.4 Operation SQA	SQA
a a c	and specified resource utilisation?	transmission load. Run the application and record				showing elapse	5.5 Mainte-	
		number of errors due to transmission failure and				time	nance	

Table 7.4.2 Resource utilisation metrics (continued)

Transmission re	Transmission resource utilization						-		
Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric	Metric	Measure	Input to	ISO/IEC	Beneficerie
	metrics		data element computations	of measured	scale	type	measurem	12207	s
				value	type		ent	SLCP	
								Reference	
Worst case	What is the absolute	What is the absolute Evaluate what is required	X = Amax / Rmax	X=>0	Absolute X =	- X	Testing	5.3	User
transmission	limit on transmission	limit on transmission for the system to reach a	Amax = $MAX(Ai)$ , (for $i = 1$ to N)	The smaller is		Count/	report	Qualifica-	
utilisation	required in fulfil a	situation of maximum	Rmax = required maximum number of	the better		Count		tion testing	Developer
	function?	load. Emulate this	transmission related error messages and				Operation	5.4	5.4
		condition. Run application failures	failures				report	Operation	Operation Maintainer
		and monitor result(s)					showing	5.5	
			MAX(Ai) = Maximum number of				elapse	Mainte-	SQA
			transmission related error messages and				time	nance	
			failures from 1st to i-th evaluation.						
			N= number of evaluations.						
Visible	What is the degree of Calibrate the test	Calibrate the test	X = SyncTime/T	The smaller	Ratio	A = size	Testing		User
synchronisation	synchronisation synchronisation	conditions. Emulate a	SyncTime = Time devoted to a continuous	the better		T = Time	report		Maintainer
	between dissimilar	condition whereby the	resource			X = ratio			SQA
	media over a set	system to reaches a	T = required time period during which				Operation		
	period of time?	situation of maximum	dissimilar media are expected to finish their	£.			report		
		transmission load. Run the	the tasks with synchronisation				showing		
		application and record the					elapse		
		delay in the processing of					time		
		different media types.							

Metric Name	Pumose	Method of application	Measurement, formula and	Interpretation Ma	stric Measure	Interpretation Metric Measure Input to ISO/IEC	Reneficeri
			data element computations	of measured sea	scale type	measure 12207	ės
				value type	2	ment SLCP	•
						Reference	
Satisfaction	How completely	Previously specify	Ratio of satisfied compliance items relating 0<= X <=1	ı	solute $A = Cou$	Absolute A= Count Specifica 5.3	Supplier
coverage of	does the software	required compliance items to efficiency	to efficiency	The closer to	B= Cor	B= Count tion of Qualifica-	:
compliance	adhere the standards,	adhere the standards, based on standards,		I is the better.	X = Cou	X= Count/ complian- tion testing []ser	g []ser
items relating	conventions or	conventions or regulations $X = 1 - (A/B)$	X = 1 - (A/B)		Count	ce and	; ;
to efficiency	regulations relating	relating to efficiency				related 6.5	
נס בוויבובווב)	to efficiency?	which to be adhered by	A= Number of failed compliance items			standards, Validation	
		software.	during testing			conven-	
						tions or	
		Design test cases in	B= Number of total compliance items			regulation	
		accordance with				S	
		compliance items.	NOTE:				
			It may be useful to collect several measured			Test	
		Conduct functional testing	Conduct functional testing values along time, to analyse the trend of			specifica-	
		for these test cases.	increasing satisfied compliance items and			tion and	
			to determine whether they are fully satisfied			report	
			or not.				

© ISO/IEC TR 9126

### 7.5 Maintainability metrics

An external maintainability metric should be able to measure such attributes as the behavior of maintainer, user, or system including the software, when the software is maintained or modified during testing or maintenance.

### 7.5.1 Analyzability metrics

An external analyzability metric should be able to measure such attributes as maintainer's or user's effort or spent resources when they try to diagnose for deficiencies or causes of failures, or for identification of parts to be modified.

### 7.5.2 Changeability metrics

An external changeability metric should be able to measure such attribute as maintainer's or user's effort by measuring the behavior of maintainer, user or system including the software when they try to implement a specified modification.

### 7.5.3 Stability metrics

An external stability metric should be able to measure attributes related to unexpected behavior of system including the software when the software is tested or operated after modification.

### 7.5.4 Testability metrics

An external testability metric should be able to measure such attributes as maintainer's or user's effort by measuring the behavior of maintainer, user or system including software when they try to test the modified or non modify software.

### 7.5.5 Compliance metrics

An external maintainability compliance metric should be able to measure an attribute such as the number of functions with, or occurrences of compliance problems, which is of the software product to failing to adhere to standards, conventions or regulations relating to maintainability which are required to be adhered.

Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric	Metric	Measure	Input to	to ISO/IEC	Beneficerie
	metrics		data element computations	of measured value	scale type	type	measurem 12207 ent SLCP Refere	12207 SLCP Reference	ø
Diagnostic function support	How many causes of failures are identified by the diagnostic function?	How many causes of Observe behavior of user failures are identified or maintainer who is by the diagnostic trying to resolve failures function?	X= A / B  A= Number of failures of which maintainer The closer to can diagnostic (using the diagnostics  1.0 is the function) to understand cause effect relation better.	0<=X<= 1 The closer to 1.0 is the better.	Absolute	A=Count B=Count X=Count/ Count	<b>1</b> · · · · · · · · · · · · · · · · · · ·	5.3 Qualifica- tion testing 5.4	5.3 Developer Qualifica- tion testing Maintainer 5.4
	Can user identify specific operation which caused failure? (User may be able to avoid falling into the same failure occurrence again with alternative operation.) Can maintainer easily find cause of failure?		B= Total number of registered failures				Operation report	Operation Operator 5.5 Mainte- nance	Operator
Data recording during operation	Can user identify specific operation which caused failure?	Observe behavior of user or maintainer who is trying to resolve failures.	Activity recording during operation ratio  X= A / B  A= Number of data actually recorded during operation  B= Number of data relating to be seconded	0<=X The closer to 1.0 is the better.	Absolute	Absolute A=Count B=Count X=Count	1	I .	Developer Maintainer
	Can maintainer casily find specific operation which caused failure?		enough to monitor status of software during operation				Operation report	Operation Operator 5.5 Mainte-nance	Operator

# Table 7.5.1 Analysability metrics (continued)

Metric Name	Purpose of the	Method of application Meast	Measurement, formula and	le	fetric Measure	Input	to ISO/IEC	Beneficerie
	metrics		data element computations	of measured savalue	scale type type	measurer	measurem 12207 ent SLCP Reference	W
Failure analysis Can user easily analyse cause of failure?  (User sometime performs maintenance by setting paramete Can maintainer easily find caus failure?  How easy to an the cause of fail NOTE: 1. It is r to represent dev 2. It is recomme measurement is presented togeth	Can user easily analyse cause of failure? (User sometimes performs maintenance by setting parameter.) Can maintainer easily find cause of failure? How easy to analyse the cause of failure? NOTE: 1. It is recomn to represent deviation. 2. It is recommended to measurement is done. presented together.	Can user easily Observe behavior of user analyse cause of remaintainer who is failure?  (User sometimes performs maintenance by setting parameter.)  Can maintainer easily find cause of failure?  How easy to analyse the cause of failure?  NOTE: 1. It is recommended to measure maxima to represent deviation.  2. It is recommended to exclude number of failur measurement is done. Though, the ratio of such presented together.	Can user easily Observe behavior of user X= Sum(T) / N analyse cause of or maintainer who is T= Tout - Tin failure?  (User sometimes performs maintenance by setting parameter.)  Can maintenance by cause of failure?  Can maintainer easily find cause of failure?  How easy to analyse the cause of failure?  NOTE: 1. It is recommended to exclude number of failures should be also measured and be used instead of time. paresented together.	0<=X Ratio X= Time/ Problem 5.3 Develope Count resolution Qualifica- The shorter is T= Time report tion testing Maintain the better.  Operation 5.4 Operator report Operation 5.5 Mainte- nance  NOTE: 3. From individual user's view, time is concerned, while effort may also concerned from maintainer's view. Therefore, person-hours may be used instead of time.	Ratio X= Time Count T= Time dual user's view. tir from maintainer's of time.	X= Time/ Problem Count resolution T= Time report Operation report report report retort s view, time is concern intainer's view. Theref	5.3 n Qualifica- tion testing n 5.4 Operation 5.5 Mainte- nance nance	5.3 Developer Qualifica- tion testing Maintainer 5.4 Operator Operation 5.5 Mainte- nance ance cd, while effort may re, person-hours may
Finding results of failure case	Can user identify specific operation which caused failure? Can maintainer easily find cause of failure?	Observe behavior of user or maintainer who is trying to resolve failures.	Failure cause finding success ratio X=1- (A/B) A= Number of failures of which causes are still not found B= Total number of registered failures	0<=X<= 1 The closer to 1.0 is the better.	Absolut A=Count c. B=Count X=Count Count	A=Count Problem B=Count resolution X=Count report Count Operation report	<b>₽ ₽</b>	5.3 Developer Qualifica- tion testing Maintainer 5.4 Operation Operator 5.5 Mainte-
Status monitoring during operation	Can user identify specific operation which caused failure? Can maintainer easily find cause of failure?	Observe behavior of user or maintainer who is trying to resolve failures.	Successful status monitoring ratio  X= 1- (A/B)  A= Number of turns which maintainer (or user) failed to get monitor data  B= Number of turns which maintainer (or user) attempted to get monitor data recording status of software during operation	0<=X<= 1 The closer to (1.0 is the better.	Absolut A=Core. e. B=Core. X=Core.	A=Count Problem B=Count resolution X=Count report Count Operation report		5.3 Developer Qualifica- tion testing Maintainer 5.4 Operation Operator 5.5 Mainte- nance

ty metrics	Method of applicat
Changeability	Purpose of the
Table 7.5.2 Cl	Metric Name

Metric Name	Dumoco of the	Mathod of application	Mascingmant formula and	Internation N	Mastric N	Manches Input			Denethouse
יייים וכי זאמוויכ	metrics	incured of application	data element computations	of measured so			measurem 12207	ļ	S
							SLCP Refere	nce	
Change	Can the user easily	Observe the behavior of	Change recording ratio X= A / B	0 <= X <= 1	A Leadur D	A= Count Problem		5.3	Developer
recordability	versions?	user or maintainer while trying to change the	A= Number of change log data actually	1.0 is the e.		S= Count resolution X= Count report		testing	Qualifica- tion testing Maintainer
		software.	recorded	better or the		Count		)	
	Can the maintainer	Otherwise, investigate	B= Number of change log data planned to	closer is 0 the		Mai	-icn-	ration	Operation Operator
	casily change the	problem resolution report	be recorded enough to trace software	less changes		ance		5.5 Mointe	
	problems?	oi mainchaile lepoit.	Citaligos	nlace tanen		າເດຕ້ອງ		-51111	
						Ope	Operation	}	
						report	E		
Ease of	Can the user or the	Observe behavior of the	Change by using parameter success ratio	0<=X<= 1		A= Count Problem	1		Developer
parameterisa-	maintainer easily	user or the maintainer	X=I-(A/B)	The closer to	Absolut I	B= Count resolution		Qualifica-	
tion	change parameter to	while trying to change the	A= Number of turns which maintainer fails			X= Count/ report	•	testing	tion testing Maintainer
	change software and	software.	to change software by using parameter	better.	_	Count			
	resolve problems?	Otherwise, investigate	B= Number of turns which maintainer			Mai	Mainten- Ope	cration	Operation Operator
		problem resolution report	attempts to change software by using			ance			
		or maintenance report.	parameter			report		Mainte-	User
							nance	Se	
						Ope	Operation		
						report	Ĕ		
Readiness for	Can the maintainer	Observe behavior of	Required effort (in person hours) to change	Ä	Ratio.	T= Time Prob	Problem 5.3		Developer
change	casily change the	maintainer who is trying	software $T = Sum(A/B)/N$	The shorter is	•	<b>Fime</b>	resolution Qua	Qualifica-	
	software to resolve	to change the software.	A= Work time spent to change	the better or		B= Size report		1 testing	tion testing Maintainer
	problem?	Otherwise, investigate	B= Changed software size	the required		N=Count			
		problem resolution report	N= Number of changes	number of		Mai	Mainten- 5.4		Operator
		or maintenance report and		changes were		ance		Operation	
		product description.	NOTE:	excessive		report	Ħ		
			1. A changed software size may be changed				5.5		
			executable statements of program code,			Ope	Operation Mainte-	inte-	
			number of changed items of requirements			report		nance	
			specification, or changed pages of						
			document etc.						

Metric Name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation Metr of measured scale value type	Metric Measure scale type type	Input measure ent	to ISO/IEC m 12207 SLCP Reference	Beneficerie s
Time spent to implement the	Can the user's problem be solved to his satisfaction	Can the user's Monitor interaction problem be solved to between user and supplier, his satisfaction Record the time taken	Time spent to update user's version  Average Time : Tav = Sum(Tu) / N	O <tav ra<="" td=""><td>Ratio Tu= 7</td><td>Time Problem resolution</td><td>3</td><td>User</td></tav>	Ratio Tu= 7	Time Problem resolution	3	User
change tor user's satisfaction	ins saustaction within an acceptable time scale?		Average 100c . (av - 5000(10) / N Tu= Trc - Tsn	the better., except of the	Irc. Tsn Time	report = Mainten-	tion testing Maintain 5.4 Operation Operator	tion testing Maintainer 5.4 Operation Operator
		ot problem.	I sn= I ime at which user finished to send request for maintenance to supplier with	number of revised		ance report	5.5 Mainte-	
			problem report.	versions was			nance	
			Tre= Time at which user received the	large.		Operation		
			revised version release (or status report)  N= Number of revised versions			report		
Time spent to	Can the maintainer	Observe the behavior of	Time to change for maintainer	0 <tav ra<="" td=""><td>Ratio Tm= 1</td><td>Tm= Time Problem</td><td>5.3</td><td>Developer</td></tav>	Ratio Tm= 1	Tm= Time Problem	5.3	Developer
implement	casily change the	the user and maintainer	Average Time: Tav = Sum(Tm) / N			resolution	Qualifica-	
change by the	software to resolve	while trying to change the	Tm=Tout - Tin	The shorter is	Tin.	report	tion testing	tion testing Maintainer
maintainer	the failure problem?	software.		the better.	Tout	11		
		Otherwise, investigate	Tout= Time at which the causes of failure	except of the	Time	Mainten-	5.4	Operator
		problem resolution report	are removed with changing the software (	number of		ance	Operation	
		or maintenance report.	or status is reported back to user)	failures was		report		
			Tin= Time at which the causes of failures	large.			5.5	
			are found out			Operation	Mainte-	
			N= Number of registered and removed			report	nance	
			failures					
	NOTE: 1. It is recon	imended to measure maxima	NOTE: 1. It is recommended to measure maximal time of the worst case and time bandwidth to represent deviation.	to represent deviat	ion.			
	2. It is recommended	<ol><li>It is recommended to exclude number of failur</li></ol>	lures of which causes are not yet found when measurement is done. Though, the ratio of such obscure failures should	neasurement is don	c. Though, th	te ratio of such	obscure failt	res should
	be also measured an	be also measured and presented together.						
	שנים חווכשאחוכת מווי	n presenteu togenier.	be also ineasured and presented together.					

Table 7.5.3 Stability metrics

Metric Name Purpose of the Method

Annual Control of the							
Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric	Measure Input	to ISO/IEC	Beneficerie
	metrics		data element computations	of measured scale	type measurem 12207	12207	s
				value type	ent	SLCP	
	To the state of th					Reference	
Less	Can user operate	Observe behavior of user	Frequency of encountering failures after	0 <x,y ratio.<="" th=""><th>A= Count Problem</th><th>5.3</th><th>Developer</th></x,y>	A= Count Problem	5.3	Developer
encountering	software system	or maintainer who is	change $X=Na/Ta$	The smaller is	T= Time resolution Oualifica-	Oualifica-	•
failures after	without failures after	without failures after operating software system	Fluctuated frequency of encountering	the better.	X= Count/ report	tion testing	tion testing Maintainer
change	maintenance?	after maintenance.	failures before/after change	The closer to	Time	9	
ı			$Y = \{(Na / Ta) / (Nb / Tb) \}$	0.0 is the	Mainten-	5.4	Operator
	Can maintainer	Count failures which user		better.	ance		
	easily mitigate	or maintainer encountered	or maintainer encountered Na = Number of turns which user		report		
	failures caused by	during operating software	encounters failures during operation after		· · · · · · · · · · · · · · · · · · ·	5.	
	maintenance side	before and after	software was changed		Operation Mainte-	Mainte-	
	effects?	maintenance.	Nb = Number of turns which user		report	nance	
			encounters failures during operation before		i.		
		Otherwise, investigate	software is changed				
		problem resolution report,	Ta = Operation time during specified				
		operation report or	observation period after software is				
		maintenance report.	changed				
			Tb = Operation time during specified				
			observation period before software is				
			changed				
	NOTE: 1. User may r	need specified period to dete	NOTE: 1. User may need specified period to determine side effects of software changes, when revision-up of software is introduced for resolving problems	in revision-up of software	e is introduced for resolu	ving problem	v
	2. It is recommend to	<ol><li>It is recommend to compare this frequency before and after change.</li></ol>	ore and after change.			0	3
	3. If changed function	is identified, it is recomme	3. If changed function is identified, it is recommended to sort out whether encountered failures are detected in the changed function itself or the any other ones. The	es are detected in the cha	noed function itself or t	he any other	The The
	extent of impacts may	extent of impacts may be rated for each failure.				ine duity outlet	JIICS, VIIIC

Table 7.5.	Table 7.5.3 Stability metrics (continued	rics (continued)							
Metric Name	Metric Name Purpose of the	Method of application	Measurement, formula and	Interpretation Metric Measure Input to ISO/IEC Beneficerie	Aetric N	deasure	Input to	ISO/IEC	Beneficerie
	metrics		data element computations	of measured scale	cale t	type	measurem 12207	12207	s
				value t	type		ent	SLCP	
								Reference	
Localisation of	Localisation of Can user operate	Count failures	Chaining failure emerging per resolved	/ X>0	Absolut /	A= Count	Absolut A= Count Problem	5.3	Developer
modification	software system	occurrences after change,	failure	The smaller c.		N= Count	resolution	N= Count resolution Qualifica-	
(Emerging	without failures after which are mutually	which are mutually	X=A/N	and the closer		X= Count/ report	report	tion testing	tion testing Maintainer
failure after	maintenance?	chaining and affected by	A= Number of failures emerged after	to 0.0 is the	_	Count			
change		change.	failure is resolved by change during	better.			Operation 5.4	5.4	Operator
(Seman)	Can maintainer		specified period				report	Operation	
	easily mitigate		N= Number of resolved failures						
	failures caused by		NOTE: It is recommend to give precise					5.5	
	maintenance side		measure by checking whether cause of					Mainte-	
	effects?		current failure is attributed to change for previous failure resolution as possible					nance	

Table 7.3.	Table 7.3.4 Testability metrics	1011103		Tack to the same And the August Augus	ļ	Massire Innit t	to ISO/IEC	Beneficerie
Metric Name	Purpose of the metrics	Method of application	Measurement. tormula and data element computations	interpretation by of measured so value ty	Metric Meas scale type type	measure cent	SLCP Reference	S
Effortless testing	Can user and Observe behavior maintainer easily or maintainer who perform operational testing software sy testing and determine after maintenance whether the software is ready to operation or not?	Observe behavior of user or maintainer who is testing software system after maintenance.	Time (effort) to test after failure resolution 0 <x and="" average="" better.="" degraded,="" exclude="" failure="" failures="" if="" is="" make="" n="Number" not="" not.="" note:="" of="" or="" reported="" resolved="" separately<="" spent="" sure="" t="Time" td="" test="" the="" them="" to="" was="" whether="" which="" x="Sum(T)"><td>er is</td><td>Ratio T= N= X= /Cou</td><td>T= Time Problem N= Count resolution X= Time report /Count Operation report</td><td></td><td>5.3 Developer Qualifica- tion testing Maintainer 5.4 Operation Operator 5.5 Mainte- nance</td></x>	er is	Ratio T= N= X= /Cou	T= Time Problem N= Count resolution X= Time report /Count Operation report		5.3 Developer Qualifica- tion testing Maintainer 5.4 Operation Operator 5.5 Mainte- nance
Readiness of built-in test function	Can user and maintainer easily perform operational testing after maintenance and get ready for operation?	Observe behavior of user or maintainer who is testing software system after maintenance.	Built-in test function use success ratio  X= A / B  A= Number of turns which maintainer can  use suitably built-in test function  B= Number of turns of test opportunities  NOTE: Examples of built-in test functions include simulation function, pre-check function for ready to use and so on.	0 < X <1 // The larger eand the closer to 1.0 is the better.	c. B C C C	A= Count Problem B= Count resolution X= Count report Count Operation report	1	
Test restartability	Can user and maintainer easily perform operational testing with checking step by step after maintenance?	Can user and Observe behavior of user maintainer easily or maintainer who is perform operational testing software system testing with checking after maintenance. step by step after maintenance?	Pause and restart of executing test run X = A / B A = Number of turns which maintainer can pause and restart executing test run at desired points to check step by step B= Number of turns of pause of executing test run	0 < X <1 The larger and the closer to 1.0 is the better.	e. B. X.	Absolut A= Count Problem e. B= Count resolution X= Count report Count Operation report	E Ē	5.3 Developer Qualifica- tion testing Maintainer 5.4 Operation Operator 5.5 Mainte- nance

5
.€
Ξ
<u>5</u>
~
Ħ
:=
25
•
2
e for mainta
÷.
-5
¥
ده
$\ddot{c}$
ব
Ξ
complia
_
ည
ت
(1)
್ಲಲ
tric
بر
7
☱
نه
empliance metr
ಡ
=
$\circ$
ᄝ
75
.5 Coi
Table 7.5.5 Co
-
0
7
7

Laure /	Compilance	אוורניון אוויס (רסוווי לאווי	Lable 1.3.3 Compilative inventes (compilative for maintainage)				
Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Ma	etric Measure	Interpretation Metric Measure Input to ISO/IEC	EC Beneficeri
	metrics	·	data element computations	of measured sca	scale type	measure 12207	7 es
				value type	8	ment SLCP	SLCP
						NCIC	
Satisfaction	How completely	Previously specify	Ratio of satisfied compliance items relating 0<= X <=1	0<= X <= 1 A	solut A= Cor	Absolut A= Count Specifica- 5.3	Supplier
coverage of	does the software	required compliance items to maintainability	s to maintainability	The closer to c.		B= Count tion of Qualifica-	fica-
compliance	adhere the standards. based on standards.	based on standards.		I is the better.	X= Con	X= Count/ complian- tion testing User	esting User
itome relating	conventions or	conventions or regulations $X = I - (A/B)$	SX = 1 - (A/B)		Count	ce and	
to	regulations relating	relating to maintainability				related 6.5	
nointeinobility		which to be adhered by	A= Number of failed compliance items			standards. Validation	lation
maintaintaint		software.	during testing			conven-	
						tions or	
		Design test cases in	B= Number of total compliance items			regulation	
		accordance with				s	
		compliance items.	NOTE:				
			It may be useful to collect several measured			Test	
		Conduct functional testin	Conduct functional testing values along time, to analyse the trend of			specifica-	
		for these test cases.	increasing satisfied compliance items and			tion and	
			to determine whether they are fully satisfied			report	
			or not.				

ISO/IEC TR 9126-2 © ISO/II

### 7.6 Portability metrics

External portability metrics should be able to measure such attribute as the behavior of the operator or system during the porting activity.

### 7.6.1 Adaptability metrics

External adaptability metric should be able to measure such attribute as the behavior of user who is trying to adapt software to different specified environments. When user has to apply an adaptation procedure other than previously provided by software for a specific adaptation need, user's effort required for adapting should be measured.

### 7.6.2 Installability metrics

External installability metrics should be able to measure such attribute as the behavior of user who is trying to install the software in a user specific environment.

### 7.6.3 Replaceability metrics

External replaceability metrics should be able to measure such attribute as the behavior of user who is trying to use the software in place of other specified software in the environment of that software.

### 7.6.4 Co-existence metrics

External co-existence ability metrics should be able to measure such attribute as the behavior of user who is trying to the software with other independent software in a common environment sharing common resources.

### 7.6.5 Compliance metrics

An external portability compliance metric should be able to measure an attribute such as the number of functions with, or occurrences of compliance problems, which is of the software product to failing to adhere to standards, conventions or regulations relating to portability which are required to be adhered.

netrics
tability m
Adapt
7.6.1
Table

.o./ arn	Lable 1.0.1 Anaplability menics	, men ics						
Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric	Metric	Measure Input to	to ISO/IEC	Beneficeric
	metrics		data element computations	of measured		type measurem 12207	12207	s
				value	type	cnt	SLCP Reference	
table data	Adaptable data Can user or	Observe user's or	Limitation free operation after data	0 <x<1< td=""><td>,</td><td>A= Count Problem</td><td></td><td>Developer</td></x<1<>	,	A= Count Problem		Developer
	maintainer easily	maintainer's behavior	adaptation		Absolut	Absolut B= Count resolution		
	adapt software to	when user is trying to	X = (A/B)	The larger	ن ن	X= Count/ report	tion testing	tion testing Maintainer
	data sets in new	adapt software to	A = The number of data which are operable and close to	and close to		Count		
	environment?	operation environment?	and not observed being incomplete	1.0 is the		Operation		Operator
			operations caused by adaptation limitations; better.	: better.		report	5.5	
			B= The number of data which is expected				Mainte-	
			to be operable in the environment to which				nance	
			the software is adapted.					
			NOTE: These data mainly include data					
			files, data tupples or databases to be					
			adapted to different data volumes, data					
			items or data structures when for example,					
			business scope is extended.					
ronmental	Environmental Can user or	Observe user's or	Adaptability to environmental business	0 <x<1< td=""><td></td><td>A= Count Problem</td><td>5.3</td><td>Developer</td></x<1<>		A= Count Problem	5.3	Developer
adaptability	maintainer easily	maintainer's behavior	environments		Absolut			
•	adapt software to	when user is trying to	X=1-(A/B)	The larger is	.;	X= Count/ report	tion testing	tion testing Maintainer
(Organization	environment?	adapt software to	A= Number of operated functions of which the better.	the better.		Count	5.4	
adaptahility to		operation environment?.	tasks were not completed or not enough			Operation		Operator
tructure of	infrastructure of Is software system		resulted to meet adequate level during			report	5.5	
in action)	enough capable to		operation testing with user's business				Mainte-	
oigainzauoii)	adapt itself to		environment				nance	
	operation		B= Total number of functions which were					
	environment?		tested.					
			NOTE: It is recommended to conduct					
			testing which takes account of varieties of					
			combination of infrastructure components of possible user's hisiness environments					
			or positive as a sounds of the sound of the					

_
(continued)
metrics (
Adaptability metrics
able 7.6.1
್ಗಡ

Laure /.o.	t Auaptaum	Table 7.0.1 Anaptability menics (commi	mcm)					
Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric		sure	Input to ISO/IEC	Beneficerie
	metrics		data element computations	of measured sa	scale	type measurem 12207	n 12207	S
				value ty	type	ent	SLCP Reference	
Environmental hardware	Can user or maintainer easily	Observe user's or maintainer's behavior	Adaptability to environmental hardwares X=1-(A/B)	Ì	bsolut	A= Count Problem B= Count resolution		Developer
adaptability	adapt software to environment?	when user is trying to adapt software to	A= Number of operated functions of which The larger is tasks were not completed or not enough the better.		ย่	X= Count report Count		tion testing Maintainer 5.4
(adaptability to hardware	Is software system	operation environment?.	resulted to meet adequate fevel during co- operating testing with environmental			Operation		Operator
devices and network	enough capable to adapt itself to operation		hardwares  B= Total number of functions which were rested				Mainte- nance	
facilities)	environment?		NOTE: It is recommended to conduct overloaded combination testing with					
			environmental hardwares which are possibly co-operated in variety user					
Environmental	Can user or	Observe user's or	Adaptability to environmental softwares	0 <x<1< td=""><td></td><td>A= Count Problem</td><td></td><td>Developer</td></x<1<>		A= Count Problem		Developer
software		maintainer's behavior	X=1-(A/B)	,	Absolut	Absolut B= Count resolution		
adaptability	adapt software to	when user is trying to	A= Number of operated functions of which The larger is		ý	X= Count/ report	tion testin	tion testing Maintainer
	environment?	adapt software to	tasks were not completed or were not	the better.		Count		5.4 Operation Operator
(adaptability to	ls software system	operation environment?.	enough resulted to meet adequate level during co-operating testing with operating			Operation		Operator
OS, network software and co-			system softwares or concurrent application				Mainte-	
operated	adapt itself to		softwares  R= Total number of functions which were				nance	
application	environment?		tested.					
software)			NOTE: It is recommended to conduct overloaded combination testing with					
			operating system softwares or concurrent					
			application softwares which are possibly					
			conviconments.					,

-	C
	Ē
	=
	⊏
٠	_
4	-
	⊏
	C
	,
	_
	_
	V
	۷
٠	' metrics
	۲.
`	۰
	Q,
	Ě
	ς
	?
•	٠.
:	-
•	_
_	c
_	_
	٠
	Ξ
	١.
	ď
,	ŕ
	•
•	⋖
	•
,	
١	/ 6 A Acantaniity
	_
Į	_
_	Ç
•	7
,	,
	Ç
•	·

LAUIC /.O.	1 Auaptaniii.	Lable 7.0.1 Anaplability inclines (confined	aca)						
Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric Measure Input to ISO/IEC Beneficerie	Metric	Measure	Input to	ISO/IEC	Beneficeric
	metrics		data element computations	of measured scale		type	measurem 12207	12207	s
				value		ı	ent	SLCP	
								Reference	
User effortless Can user or	Can user or	Observe user's or	User effort required to adapt to user's	0 <t.< td=""><td>Ratio</td><td>T=Time</td><td>Ratio T=Time Problem 5.3</td><td></td><td>Developer</td></t.<>	Ratio	T=Time	Ratio T=Time Problem 5.3		Developer
adantation	maintainer easily	maintainer's behavior		er is			resolution Qualifica-	Qualifica-	
acab carro	adapt software to	when user is trying to	T=Sum of user operating time to be spared the better.	the better.			report	tion testing	tion testing Maintainer
	environment?	adapt software to	to complete adaptation of the software to					5.4	
		operation environment?.	user's environment, when user attempt to				Operation	Operation Operation Operator	Operator
			install or change setup.				report	5.5	
			(Person-hour may be used instead of time.)					Mainte-	
								nance	

Metric Name								
	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric		Input to	Measure Input to ISO/IEC	Beneficerie
	metrics	•	data element computations	of measured scale		measurem 12207	12207	s
				value type		ent	SLCP	
				•			Reference	
Easiness of	Can user or	Observe user's or	X = 1 - (A/B)	0<=X<= 1 Absc	Absolut A=Count Problem	Problem	5.3	Developer
Setun Re-try	maintainer easily re-	maintainer easily re- maintainer's behavior	A = Number of turns which user fail to re-		B=Count	B=Count resolution		
,	try setup installation	try setup installation when user is trying to re-	try setup during setup operation	1.0 is the	X=Count/ report		tion testing Maintainer	Maintainer
	of software?	try setup installation of	B = Total number of turns which user	better.	Count			
		software?	attempt to re-try setup during setup			Operation	Operation	Operator
			operation			report	5.5	
			•				Mainte-	
			NOTE: 1. This metric is suggested as				nance	
			experimental use.					
Operational	Can user or	Observe user's or	X = A/B	0<=X<= 1 Abs	Absolut A=Count Problem	Problem	5.3	Developer
installation	maintainer easily	maintainer's behavior	A = Number of turns which a user	The closer to e.	B=Count	B=Count resolution	Qualifica-	
flexibility	install software to	when user is trying to	succeeded to change install operation for	1.0 is the	X=Count/ report	report	tion testing	tion testing Maintainer
famous and	operation	install software to	his/her convenience	better.	Count		5.4	
	environment?	operation environment	B = Total number of turns which a user			Operation	Operation Operator	Operator
		•	attempted to change install operation for			report	5.5	
			his/her convenience				Mainte-	
							nance	
			NOTE: 1. This metric is suggested as					
			experimental use.					

ISO/IEC TR 9126-2

Table 7.6.2 Installability metrics (continued)	
NOTE: The following complementary metrics may be used.	3) Operational installation effort reduction User Install Operation Procedure Reduction Ratio X = 1- (A / B) A = Number of install operation procedures which a user had to do after reduced
1) Effortless installation User manual actions for installation $X=A$	B = Number of install operation procedures normally $0 < X < I$
A= The number of user manual actions for installation 0 <x< td=""><td>The closer to 1.0 is the better.</td></x<>	The closer to 1.0 is the better.
The smaller is the better.	4) Easiness of user's manual install operation Easiness level of user's manual install operation
2) Installation easiness Installation supporting level $X = A$	S = Score of easiness level of user's manual operation Examples of easiness level are following:
A is rated with, for example:	[very easy] only user's watching except just start install or setup functions:
<ul> <li>Just only executing installation program and nothing more is needed (excellent):</li> <li>Instucting function guide for installation (good);</li> </ul>	[casy] only user's answering to question from install or setup functions; [not so easy] user's looking up parameters from tables or fill-in-boxes to be changed and
<ul> <li>Source code of program is needed to be modified for installation (poor).</li> <li>X= Direct Interpretation of measured value</li> </ul>	setting them: [complicated] user's seeking parameter files, looking up parameters from files to be changed and writing them.

X= Direct Interpretation of measured value

		2) 4 4 4 4
	2	4
	<u> </u>	
	300	
,	2	3
6	ž	
,	2	
,	4	)
•	Č	d

Table 7.6.	Table 7.6.3 Replaceability metrics	ity metrics						- 1	
Metric Name	Purpose of the metrics	Method of application	Measurement, formula and data clement computations	Interpretation Metric of measured scale value type	Metric scale type	Measure Inputype mea	Input to ISO/IEC measurem 12207 ent SLCP Reference		Beneficerie s
Data continuation	Can user or maintainer easily continue to use the same data after replacing this software to previous one?  Is software system migration going on successfully?	Observe user's or maintainer's behavior when user is replacing software to previous one?	Data continuously use ratio $X = A / B$ $0 <= X <= 1$ A = number of data which are used in other. The larger is software to be replaced and are confirmed the better. that they are able to be continuously used. B = number of data which are used in other software to be replaced and planned to be continuously reusable.	0<= X <= 1 The larger is the better.		Absolut A= Count Problem 5.3 e. B= Count resolution Qualifica- X= Count report tion testing Count Operation Operation report 5.5 Mainte- nance	Ë Ë	Di fica- esting M ation O tc-	5.3 Developer Qualifica- tion testing Maintainer 5.4 Operation Operator 5.5 Mainte- nance
Function	Can user or maintainer easily continue to use similar functions after replacing this software to previous one?  Is software system migration going on successfully?	Observe user's or maintainer's behavior when user is replacing software to previous one?	Function inclusion ratio X = A / B 0<= X <= 1 A = number of functions which produce as The larger is enough similar results as used to be the better. produced and of which changes have not to be required; B = number of tested functions which are similar to functions provided by other software to be replaced.	0<= X <=1 The larger is the better.	i	Absolut A= Count Problem 5.3 Developer c. B= Count resolution Qualifica- X= Count/ report tion testing Maintainer Count 5.4 Operation Operation Operator report 5.5 Mainte- nance	Problem 5.3 resolution Qualific report tion testi 5.4 Operation Operati report 5.5 Mainte- nance	fifica- cesting N attion C nte-	5.3 Developer 1 Qualifica- tion testing Maintainer 5.4 1 Operation Operator 5.5 Mainte- nance

## Table 7.6.4 Co-existence metrics

TADIC /.V	TABLE 1.0.4 CO-CAISICILICE INCUINCE	c iliculus							
Metric Name	Metric Name Purpose of the	Method of application	Measurement, formula and	Interpretation Metric Measure Input to ISO/IEC Beneficerie	Metric	Measure	Input to	ISO/IEC	Beneficerie
	metrics		data element computations	of measured scale	scale	type	measurem 12207	12207	s
				value	type		ent	SLCP	
								Reference	
Concurrent	How often user	Use concurrently the	Concurrent multiple software usable ratio $0 \le X \le 1$	0<=X<= 1	Ratio	A= Count Problem 5.3	Problem		Developer
multiple	encounter any	g	X = A/T	The closer to		B= Count	resolution	B= Count resolution Qualifica-	
software use	constraints or	and several other software	and several other software A = Number of any constraints or	0 is the better.		X= Count/ report	report	tion testing Maintainer	Maintainer
with loce	unexpected failures	which user often use.	unexpected failures which user encounter			Count		5.4	
constraints	when user operate		during operating concurrently other				Operation	Operation Operation SQA	SQA
College Alline	concurrently other		software				report	5.5	
	software?		T = Time to operate concurrently other					Mainte-	Operator
			software					nance	

$\subseteq$
4
Ξ
7
[2]
<u></u>
00rt
<u></u>
.0
foj
compliance
=
.53
_
=
0
્ပ
_
~3
ric
ہ۔
40
ē
₽
empliance metric
ပ္
=
.2
=
- 22
8
6
$(\tilde{})$
.5 C
W
Ġ
$\overline{}$
(
9
$\overline{\mathbf{c}}$
7

Metric Name	Purpose of the	Method of application	Measurement, formula and	Interpretation Metric	Interpretation Metric Measure Innut to ISO/IEC	/IEC Reneficeri
	metrics	:	data element computations	of measured scale	type measure 12207	07 es
				value type	ment SLUF Refere	Seference
Satisfaction	How completely	Previously specify	X = 1 - (A/B)	0<= X <=1	A= Count Specifica- 5.3	Supplier
coverage of	does the software	required compliance items		The closer to Absolut	B= Count tion of Que	difica-
compliance	adhere the standards, based on standards,	based on standards.	A= Number of failed compliance items	1 is the better. e.	1 is the better. e. X= Count/ complian- tion testing User	testing User
items relating	conventions or	conventions or regulations during testing	during testing		Count ce and	ı
to portability	regulations relating	relating to portability			related 6.5	
	to portability?	which to be adhered by	B= Number of total compliance items		standards, Validation	idation
		software.			conven-	
		Design test cases in	NOTE:		tions or	
		accordance with	It may be useful to collect several measured		regula-	
		compliance items.	values along time, to analyse the trend of		tions	
		Conduct functional testing	Conduct functional testing increasing satisfied compliance items and		Test	
		for these test cases.	to determine whether they are fully satisfied		specifica-	
			or not.		tion and	
					report	

### Annex A (Informative) Descriptions of the metrics tables

The purpose of these descriptions is to clarify the reader comprehension for the quality metrics formulas proposal in the clause 7 in the technical reports 9126-2, 3 and 4.

### A.1 Metrics name

Metrics name characterizes measureable attribute of software and represents a unique or group of measurements.

Metrics name has the same or similar name in internal, external and quality in use metrics, when they are intended to be mutually corresponded respectively.

### A.2 Purpose of the metrics

This helps to identify what user of metric can know by using the metric.

NOTE: This is described as a questionary style to look up easily in accordance with Goal / Question / Metric flamework.

### A.3 Method of application

This helps to understand what way are useful and recommended to apply metrics.

### A.4 Measurement, formula and data element computations

This helps to understand what kind of measurement, formula and data element are used to compute measure.

### A.5 Interpretation of measured value

This helps to understand the range of measured value and the interpreted better range.

ISO/IEC TR 9126-2

#### A.6 Metric scale types

The following measurement metric scale types should be identified for each measure, when a user of metrics has the result of a measurement and uses the measure for calculation or comparision. The average, ratio or difference values may have no meaning for some measures. Such Metric scale types are: Nominal scale, Ordinal scale, Intervals scale, Ratio scale, Absolute scale. M'=F(M), where F is the admissible function, explain what the admissible function is (if M is a metric then M'=F(M) is also a metric).

#### a) Nominal scale

M'=F(M) where F is any one-to-one mapping.

This includes classification, for example, software fault types (data, control, other). An average has a meaning only if it is calculated with frequency of the same type. A ratio has a meaning only when it is calculated with frequency of each mapped type. Therefore, the ratio and average may be used to represent a difference in frequency of only the same type between early and later cases or two similar cases. Otherwise, they may be used to compare mutually frequency of each other type respectively.

#### b) Ordinal scale

M'=F(M) where F is any monotonic increasing mapping that is, M(x)>=M(y) implies M'(x)>=M'(y).

This includes ordering, for example, software failure by severity (negligible, marginal, critical, catastrophic). An average has a meaning only if it is calculated with frequency of the same mapped order. A ratio has a meaning only when it is calculated with frequency of each mapped order. Therefore, the ratio and average may be used to represent a difference in frequency of only the same order between early and later cases or two similar cases. Otherwise, they may be used to compare mutual frequency of each order.

#### c) Intervals scale

M'=aM+b (a>0)

This includes artificial rating scales, for example, rating scales of sensitive questionnaire for asking about usability. These rating scales are comparable and an average has meaning only if it is calculated with the same rated scales for the same asking, but a ratio of such rating scales has no meaning. Even when double scores, it is pointed out only that a score is twice as much as the another, but not that anything is double.

#### a) Ratio scale

M'=aM (a>0)

This includes time interval, for example, time between software failures occurred. An average and a ratio has meaning respectively and they give actual meaning to the values. When the value is double, it is pointed out it takes twice as long as the another.

## e) Absolute scale

M'=M

they can be measured only in one way.

As an appropriate statistics, Mode and Frequency can be used for Nominal type, Median and Percentile for Ordinal type, Mean and Standard deviation for Interval type, and Geometric mean and Coefficient of variation for Ratio type. All arithmetic analysis of resulting count is meaningful for Absolute type.

For examples, line of code is Ratio type, because there are many different way to measure it( such as LOC, number of characters, and number of bytes). A count of the number of failure and the number of people working on a software project are Absolute type, because they can be measured only in one way.

© ISO/IEC ISO/IEC TR 9126-2

#### A.7 Measurements types

For designing procedure for collecting data, interpreting fair meanings, and normalizing measures for comparison, a user of metrics should identify and take account of measure type of measurement employed by a metric.

#### A.7.1.Size measure type

A measure of this type represents a particular size of software according what it claims to measure within its definition.

NOTE: that software may have many representations of size ( like any entity can be measured in more than one dimension - mass, volume, surface area etc.).

Normalizing other measures with a size measure can give comparable values in terms of units of size. The size measures in this report can be used for software quality metrics.

#### A.7.1.1 Functional size type

Functional size is an example of one type of size (one dimension) that software may have. Any one instance of software may have more than one functional size depending on, for example:

- the purpose for measuring the software size (It influences the scope of the software included in the measurement);
- the particular functional sizing method used (It will change the units and scale).

The definition of the concepts and process for applying a functional size measurement method (FSM Method) is provided by the standard ISO/IEC 14143-Part1.

In order to use Functional Size for normalization a quality assessor needs to ensure that the same functional sizing method is used and that the different software being compared have been measured for the same purpose and consequently have a comparable scope.

Although the following often claim that they represent functional sizes, it is not guaranteed they are equivalent to the functional size obtained from applying an FSM Method and compliant with ISO/IEC 14143-Part1. However, they are widely used in software development:

- 1. number of spread sheets;
- 2. number of screens;
- 3. number of files or data sets which are processed;
- 4. number of itemized functional requirements described in user requirements specifications.

#### A.7.1.2 Program size type

In this clause, the term 'programming' represents the number of executions resulting in an action, and the term 'language' represents the type of expression used.

#### 1) Program source size

Programming language should be explained and it should be provided how the non executable statements, such as comment lines, are treated. The following measures is commonly usable.

a) Non-comment source statements (NCSS)

Non-comment source statements (NCSS) include executable statements and data declaration statements with logical source statements.

#### b) New program size

A developer may use newly developed program size to represent development and maintenance work product size.

#### c) Language use

For a same executable statement, the program size depend on the language use.

d)Changed program size A developer may use changed program size to represent size of software containing modified components.

NOTE: Example of computed program size formula is: new lines of code + 0.2 x lines of code in modified components (NASA Goddard).

It may be needed to distinguish a type of statements of source code into more detail like followings.

#### i) Statement type

- Logical Source Statement(LSS). The LSS measures the number of software instructions. The statements are irrespective of its relationship to lines and independent of the physical format in which they appear.
- Physical Source Statement(PSS) The PSS measures the number of software source lines of code.

#### ii) Statement attribute

- -Executable statements;
- -Data declaration statements;
- Compiler directive statements;
- Comment source statements.

#### iii) Origin

- Modified source statements;
- Added source statements;
- Removed source statements;
  - Newly Developed source statements: (= added source statements + modified source statements);
  - Reused source statements: (= original modified removed source statements);

© ISO/IEC TR 9126-2

## 2) Program word count size

The measurement may be computed by the following so-called Halstead's measure:

Program vocabulary = n1+n2; Observed program length = N1+N2, where is:

- n1: Number of distinct operator words which are prepared and reserved by program language in a program source code;
- n2: Number of distinct operand words which are defined by programmer in a program source code;
- N1: Number of occurrences of distinct operators in a program source code;
- N2: Number of occurrences of distinct operands in a program source code.

#### 3) Number of modules

The measurement is counting the number of modules of a program.

## A.7.1.3 Utilized resource size measure type

This type identifies resources utilized by the operation of the software being evaluated. Examples are:

- a) Amount of memory ex.) amount of disk memory occupied temporally or stable during the software execution;
- b) I/O load ex.) bit size of communication data (meaningful for backup tools on a network);
- c) CPU load ex.) percentage of occupied CPU instruction sets per second (meaningful for CPU utilization and efficiency of process distribution in multi-thread software's running on concurrent/parallel systems);
- d) Files and data records ex.) bit size of file or record;
- e) Documents ex.) number of document pages.

It may be important taking note of peak (maximal) and average values, as well as periods of time and number of observations done.

## A.7.1.4 Specified operating procedure step type

This type identifies static steps of procedure which are specified in a human-interface design specification or a user manual.

The measured value may differ depending on what kinds of description are used for measurement, such as a diagram or a text representing user operating procedure.

ISO/IEC TR 9126-2

## A.7.2 Time measure type

The user of metrics of time measure type should record time periods, how many examined sites and how many users took part of measurements.

The user of metrics should be aware that there are many ways in which time can be measured as a unit, including following:

#### a) Real time unit

This is physical time: i.e. second, minute, or hour. This unit is usually used for describing task processing time of real time software.

## b) Computer machinery time unit

This is computer processor's clock time: i.e. second, minute, or hour of CPU time.

## c) Official scheduled time unit

This includes working hours, calendar days, months or years.

## d) Component time unit

When there are multiple sites, component time identifies individual site and it is an accumulation of individual time of each site. This unit is usually used for describing component reliability, for example, component failure rate.

#### e) System time unit

When there are multiple sites, system time does not identify individual site but all the sites running, because whole sites are involved into one system. This unit is usually used for describing system reliability, for example, system failure rate.

### A.7.2.1 System operation time type

System operation time type provides a basis for measuring time of software availability. This is mainly used for reliability evaluation. It should be identified whether the software is under discontinuous operation or continuous operation. If the oftware operates discontinuously, it should be assured that time measurement is done just on the periods the software is active (this is obviously extended to continuous operation).

#### a) Elapsed time

When use is constant, for example in systems operating for the same length of time each week.

### b) Machine powered-on time

For real time, embedded or operating system software that is in full use the whole time the system is operational.

#### c) Normalized machine time

As in "machine powered-on time", but pooling data from several machines of different power and applying a correction factor.

#### A.7.2.2 Execution time type

Execution time type is the time which is needed to execute software to complete a specified task. The distribution of several attempts should be analyzed and mean, deviation or maximal should be computed. The execution under the specific conditions, particularly overloaded condition, should be examined. Execution time measure type is mainly used for efficiency evaluation.

© ISO/IEC TR 9126-2

#### A.7.2.3 User time type

User time type is measured upon time periods spent by individual user on completing tasks by using operations of the software. Some examples are:

#### a) Session time

Measured between start and end of a session. Useful, as example, for drawing behavior of users of a home banking system. For an interactive program where idling time is of no interest or where interactive usability problems only are to be studied.

## b) User operating time ( task time )

Time spent by an individual user to accomplish a task by using operations of the software at an each attempt. It should be well defined what will be the start and end points of the measurement.

#### c) User time

Time spent by an individual user to use the software from the getting started to now. (Approximately, it is how many hours or days user uses the software from beginning.)

#### A.7.2.4 Effort type

Effort type is the productive time associated with a specific project task.

#### a) Individual effort

This is the productive time which is needed for the individual person who is developer, maintainer, or operator to work to complete a specified task. Individual effort assumes productive hours only according to a certain number of productive hour per day.

#### b) Task effort

Task effort measure is an accumulated value of individual for all the project personnel: developer, maintainer, operator, user or others who worked to complete a specified task.

#### A.7.2.5 Time interval of events types

This measure type is the time interval between event and next one during observation time period. The frequency with observation time period may be used in place of this measure. This is typically used for describing the time between failures occurring successively.

#### A.7.3 Count measure type

This measure type identifies number of counted number, turn, event or incident with investigation activity.

Investigation activity includes reviewing, testing and operating, and using from view of human-engineering and ergonomics. This measure type should be attached to informative description on the context including periods, and number and profile of experimented site and users during counting are performed, because the measure strongly depends on those.

If attributes of documents or software product are counted, they are static count type. If events or human actions are counted, they are kinetic count type. The followings are belong to static count type.

ISO/IEC TR 9126-2

#### A.7.3.1 Number of detected fault type

The measurement is to count the detected faults during reviewing, testing, correcting, operating or maintaining. Severity level may be used to categorize them for taking account of impacts.

## A.7.3.2 Program structural complexity number type

The measurement is to count program structural complexity. Examples are number of distinct paths or McCabe's cyclomatic number.

## A.7.3.3 Number of detected inconsistency type

This measure is to count detected inconsistent items which are prepared for investigation.

## a)Number of failed conforming items

#### Examples:

- Conformance to specified items of requirements specifications;
- Conformance to rule, regulation, or standard;
- Conformance to protocol, data format, media format, character code, or other.

## b)Number of failed user expectation

The measurement is to count satisfied/unsatisfied list items, which describe gaps between user's reasonable expectation and software product performance.

The measurement use a questionnaires for asking tester, customer, operator, or end user which kinds of deficiencies are discovered.

The followings are example items for specified user callable function:

- Actually available or not;
- Actually operable effectively or not;
- Actually operable to user's specific intended use;
- Actually Expected/needed or not.

## A.7.3.4 Number of change type

This type identifies software configuration items which are detected they have been changed. Example is number of changed places in source code.

NOTE: Changed source lines of code may be counted as changed places in source code.

The followings are belong to kinetic count type.

#### A.7.3.5 Number of detected failure type

The measurement is to count the detected failures during product development, testing, operating or maintaining. Severity level may be used to categorize them for taking account of impacts.

© ISO/IEC ISO/IEC TR 9126-2

#### A.7.3.6 Number of attempt (trial) type

This measure is to count intended operation action of user, checking and finding action of personel, questionnaires and replies, or test cases for investigation during reviewing, testing, correcting, operating or maintaining. Examples are number of detection of design anomalies during the review, number of misunderstanding requirements at the specification level, number of actually tried test cases, number of cancellation operation and so on.

## A.7.3.7 Stroke of human operating procedure type

This measure is to count strokes of user human action as kinetic steps of procedure during user is interactively operating the software. This measure quantifies an ergonomic usability as well as an effort to use. Therefore, this is used for usability metrics. Examples are number of strokes to perform a task, number of eye movements, and so on.

#### A.7.3.8 Score type

This type identifies the score or arithmetic calculation result. Score may includes count and calculation of checking on/off on check lists. Examples: Score of check list; score of questionnaire; Delphi method; etc.

#### A.8 Input to measurement

This helps to understand what kinds of information or documents are generally required to do measurement.

#### A,9 ISO/IEC 12207 SLCP Reference

This suggest processes of software life-cycle processes (SLCP) defined in ISO/IEC 12207, in which the metric is beneficially applicable.

#### A.10 Beneficeries

This helps to understand whose view and benefit are strogly related with the metric, though any other personel and party related to the software are also receive benefit.

# Annex B (Informative) Remarks for better use of metrics

#### B.1 Take accounts of constraints of applied metrics

The measures may not be interpret adequately and not understood sufficiently without context describing situation and condition during data are gathered and metrics are applied.

For examples, 1)"time required to learn operation" measure is often different between skillful operators in similar software systems and unskillful operators. 2) If testing is not so much intensive, measures acquired as testing result may be different so much from the true value.

Therefore, it is recommended to take accounts the following contexts to avoid to interpret inadequately and to misunderstand the measures of metrics.

## a) Differences between testing environment and actual user operation one

Are there any remarkable differences between testing environment and actual user operation one?

The followings are examples:

- -testing with higher / comparable / lower performance of CPU of users' computer;
- -testing with higher / comparable / lower performance of network and communication;
- -testing with higher / comparable / lower performance of operating system;
- -testing with higher / comparable / lower performance of user interface type provided by operating system.

## b) Differences between testing execution and actual user operational execution

Are there any remarkable difference between testing execution and actual user operational execution?

The followings are examples:

- -coverage of functional specification or program by test cases;
- -test case sampling ratio;
- -high speed real time processing testing;
- -over loaded data processing testing;
- -non-stop operation testing;
- -abnormal, exception, or fault injected data input testing;
- -frequency of use, such are daily, weekly, monthly, or only at emergency;
- -combination of considerable other software, devices, equipment, and systems.

© ISO/IEC TR 9126-2

#### c) User profile under observation

Are there any remarkable different user profile between tested and actual?

The followings are examples:

- -heavy, moderate, less or temporal user;
- -trained level, e.g., skillful operator or first entry operator;
- -expert user or ordinal user;
- -office user or home user.

#### d) Data validation level

Are there any problems come from what kinds of data collection procedures and level of data validation?

The followings are examples.

#### -procedures of data collection:

automatically with tools or facilities/ manually collected / questionnaires or interviews;

#### -data source report level:

developers' self reports / reviewers' report / inspected report by evaluator;

#### -data validation activity:

developers' self check / inspection by independent evaluators.

#### e) Balance of the extent of the investigation performance and measures

Are there remarkable problems in investigation performance?

It is important to take consider to keep balance between the extent of the investigation performance and measures (detected results) in fair range during testing;

Investigation includes reviewing and testing performance.

Most of the external metrics use measurement value derived from testing cases and results of problem detection in a validation testing or operation testing. Most of internal metrics also use derived from review. Therefore, the measured value of external metrics are affected by the extent of the testing effort, that is, investigation performance. Internal metrics also may be similarly affected the extent of the reviewing, that is, investigation performance.

The user of metrics should identify the extent of justification of the measured values depends on the extent of the investigation performance.

The extent of investigation to detect problems as input affects the extent of detected problems (failures, faults, miss-matching, etc.) and the extent of residual covered problems as outputs.

## f) Balance of the extent of the specification clearance and conformance testing

Are there remarkable problems in specification clearance?

Metrics often use measurements which count problems by comparing specification with testing results and by determining whether they are consistently matching or not, that is, whether the software conform to its specification or not. However, specification may not be enough mature to do that appropriately. Therefore, it is recommended to conduct to review and to improve the specification from a view of software product fulfilling its specific intended use in actual, during specification based matching testing case are designed and tested.

## B.2 Validity demonstration and use of metrics

The user of metrics should identify the methods for metrics validity demonstration, which are following below.

#### a) Correlation

The variation in the quality characteristics values ( the measures of principal metrics in operational use ) explained by the variation in the metric values, which is given by the square of the linear coefficient.

An evaluator can predict quality characteristics without measure directly by using these metrics which have correlation ability.

#### b) Tracking

If a metric M is directly related to a quality characteristics values Q (the measures of principal metrics in operational use), for a given product or process, then a change value Q(T1) to Q(T2), would be accompanied by a change metric value from M(T1) to M(T2), which is the same direction (for example, if Q increase, M increase).

An evaluator can detect movement of quality characteristics along time preriod without measure directly by using these metrics which have tracking ability.

#### c) Consistency

If quality characteristics values ( the measures of principal metrics in operational use ) Q1, Q2,..., Qn, corresponding to products or process 1, 2,..., n, have the relationship Q1 > Q2 > ..., Qn, then the correspond metric values would have the relationship M1 > M2 > ..., Mn.

An evaluator can notice exceptional and error prone components of software by using these metrics which have consistent ability.

#### d) Predictability

If a metric is used at time T1 to predict a quality characteristics values Q (the measures of principal metrics in operational use at T2, prediction error, which is { (predicted Q(T2)- actual Q(T2)) / actual Q(T2)}, would be with allowed prediction error.

An evaluator can predict the movement of quality characteristics in the future by using these metrics which have predictability.

#### e) Discriminative

A metric would be able to discriminate between high quality software components and low quality software components.

An evaluator can categorize software components and rate quality characteristics values by using these metrics which have discriminative ability.

#### **B.3** Prediction use of metrics

Early detection and prediction of quality of the software product is a one of the most effective use of metrics.

#### a) Future prediction

## - Future measure prediction

To estimate the future values of the same measure by using the current measured values, it is estimated based on trend along with time.

For example, the measured value trend of mean time between failures during testing can be used to estimate the value of mean time in actual operation.

#### - Future invisible other measure prediction

To estimate the future values of the invisible other measure by using the current measured values, it is estimated based on correlative relations.

For example, the complexity of modules during coding may be used to predict time or effort of program change and test during maintenance.

#### b) Current fact finding prediction

#### - Invisible other measure prediction

To estimate the current values of other measure which is supposed to be strongly related mutually by using the current measured values, it is estimated based on correlative relations.

For example, because the number of remaining faults in a software product is an not measurable, it may be estimated by using the number and trend of detected faults.

The metrics which are designed for predicting the attributes should be documented with explanations including below:

- -models for predicting the attribute;
- -formula for predicting the attribute;
- -experience for predicting the attribute;
- -justification for predicting the attribute.

The metrics which are designed for predicting the attributes may be sophisticated with the metrics validation procedure containing below:

- -identify the samples of measures of attributes which are to be predicted
- -identify the metrics which are supposed to be capable for prediction
- -perform a statistical analysis
- -document the results
- -iterate above periodically

#### **B.4 Detect quality problem prone components**

The following methods are usually employed to detect quality problem prone components:

- -extremely deviated in distribution;
- -exceeding boundaries of adequate pair of upper or lower limit in trend;
- -extremely deviated or exceeding boundaries of adequate pair of upper or lower limits correlation.

It is recommended to use chrats such Pareto, trend along the time or histograms.

#### **B.5** Displaying measurement results

#### a) Displaying quality characteristics evaluation result

For example, following graphical presentations are useful to display quality evaluation result for each quality characteristics and subcharacteristics:

Radar chart; Bar chart and so on.

#### b) Displaying measures

They are useful grafical presentations such Pareto chart, trend chart along the time, histograms, correlation chart and so on.

© ISO/IEC ISO/IEC TR 9126-2

## **Annex C (Informative)**

# Example of Quality Measurement in the industry using ISO/IEC 9126 concepts

This clause presents a step by step quality approach using ISO/IEC 9126 concepts and quality metrics. This example is a summary of the steps use to put in place, execute a quality process and measure the software quality. This example is based on five-year experience and implementation in the industry. The objective of this clause is to procure to the reader an idea on how he could implement a quality process using ISO9000 and ISO/IEC 9126 concepts in the development process.

#### C.1 Overview of development and quality process

#### a) Development process view for software

These figure present an overview of the software life cycle process, quality approach steps and quality metrics (Internal, external and quality in use) at each step of the development process. At this stage, Quality criteria and internal quality and metrics are use to evaluate the software quality.

	Step #1	Step #2	Step #3	Step #4
Development Process (ISO/IEC 12207 Software Life-Cycle Processes)	Software Requirements Analysis	Software Architectural Design	Software Detailed Design	Software Coding and Testing (Coding)
Quality Approach and Quality Deliverables	-Goal Quality (GO) -Required Product Quality (RPQ) -Quality Criteria and Weight -Quality Strategy Plan -Testing Strategy Plan -Configuration Plan	-Design Quality (DQ)  -Development methodology  -Standards  -Design standards  -Program structure	-Estimated / Predicted Product Quality -Internal Quality Characteristics and Measurement	-Estimated / Predicted Product Quality -Internal Quality Characteristics and Measurement

Figure C.1a Quality Approach related to Development Process of Software Life Cycle Process

# b) Software qualification Testing process for software

At this stage, external metric is use to evaluated the software quality.

Step #5	Step #6	Step #7	Step #8	Step #9
Software Coding and Testing  (Unit Test)	Software Integration,  Software Qualification Testing (Integrated Testing)	System Integration,  System Qualification Testing (System Testing)	Software Acceptance Support (Acceptance Testing)	Software Installation Support, Operational Testing
-Testing Quality Quality Measure -Incident Report	-External Quality Characteristics and Measurement -Testing log -Test Incident Report -Test Summary Report -Pass/Fail Criteria -Signoff	-Complete Functionality  -Testing Condition:  1) Normal  2) Exceptional  3) Stress and Volume  -Quality -External Quality Characteristics and Measurement  -Test log  -Test Incident Report  -Pass/Fail Criteria  -Signoff	-Complete Functionality  -Performance Testing  -Quality -External Quality Characteristics and Measurement  -Test log  -Test Incident Report  -Pass/Fail Criteria -Signoff	-External Quality, Quality In Use Characteristics and Measurement  -Test log  -Test Incident Report

Figure C.1b Quality approach related to Development process of Software Life cycle Process

#### C.2 Quality Approach Steps

Step #1 Goal Quality

Software Requirements analysis represent the necessary requirements from the real user. Goal Quality (GQ) represents the necessary and sufficient quality reflecting real user needs. The following relative weights for each quality characteristics have been assigned, based on the quality goal, objectives and the requirement (present at the software requirement analysis and quality requirement). Assigning relative weights will allow the developers and users to focus their efforts on the most important aspects of the software /system. The Quality Strategy Plan will describe the quality requirement, measurement and signoff (pass/fail quality criteria) process for each step of the software development process. The testing strategy plan will focus on the certification process using quality characteristic, subcharacteristic and measuring defect.

Table C.2.1 Example of overall quality characteristics and weights

CHARACTERISTIC	WEIGHT
	(High/Medium/Low)
Functionality	
Reliability	
Usability	
Efficiency	
Maintainability	
Portability	
Quality in Use	

ISO/IEC TR 9126-2

Table C.2.2 Example of overview of quality subcharacteristics

Characteristic	Subcharacteristic	User WEIGHT	Technical WEIGHT	Goal Quality Required Product Quality
Functionality	Suitability			
	Accuracy	***************************************		
h	Interoperability			
	Compliance			
WINDOW P. L.	Security			
Reliability	Maturity (hardware/software/data)			
	Fault tolerance			
	Recoverability (data, process, technology)			
	Redundancy			
<sup>'</sup> sability	Understandability			
	Learnability			
	Operability			
	Attractiveness			
Efficiency	Time behavior			
	Resource utilisation			
Maintainability	Analyzability			
	Changeability			
	Stability			
	Testability			The second secon
Portability	Adaptability			
	Installability			
THE OTHER PROPERTY.	Replaceability			
	Co-existence			
Quality In Use	Effectiveness			
	Productivity			
	Safety			
· · · · · · · · · · · · · · · · · · ·	Satisfaction			

## Step #2: Design Conformity

All the next steps are driven by the quality criteria and characteristic, subcharacteristic weight.

Step #2 identify the conformity for Data, Process and Technology, using metrics. For each step, a pass or fail criteria are verify using metrics.

QUALITY ASSURANCE FORM			NCE FORM		No	: 1	
Design Phase (1/1) Charac. Sub-Char.		Proces Proces Archi			Date: _		
		Indicators to measure			Value	Freq.	Resp.
F u n	Suitability and Accuracy	FSu-1a	Process specification change ratio	NB of change to a process due to mis-understanding of the requirements  NB Total process		after reviews and at each change request	QA Team
t i o n		FSu-1b	Process specification change ratio	Nb of change to process due to change in environments(legal, organizational, technological)  Nb Total Process		after reviews and at each change request	QA Team
a l i		FSu-1c	Process specification change ratio	Nb of change to Process due to change in the requirements  Nb Total Process		after reviews and at each change request	QA Team
y							

Figure C.2.1a An example of Metrics Form Sheet

## Step #3: Software Detailed Design Conformity

Step #3 identify the requirements conformity for data, function and technology, using metrics.

П		QUALITY ASSURANCE FORM					: 1
Software Detailed Design Phase (1/1)		Funct Funct Analy		_ Date: _		-	
Charac. Sub-Char.		Indicators to measure			Value	Freq.	Resp.
F u n	Suitability and Accuracy	FSu-1a	Functional specification change ratio	NB of change to functions due to mis-understanding of the requirements  NB Total Function		after reviews and at each change request	QA Team
t i o n		FSu-1b	Functional specification change ratio	Nb of change to functions due to change in environments(legal, organizational, technological)  Nb Total Function		after reviews and at each change request	QA Team
a l i		FSu-1c	Functional specification change ratio	Nb of change to functions due to change in the requirements  Nb Total Function		after reviews and at each change request	QA Team
<u>y</u>							

Figure C.2.1b An example of Metrics Form Sheet

## Step #4: Software Coding Conformity

Step #4 identify the conformity for coding using internal metrics.

		QUALITY ASSURANCE FORM				No : 1	
Software Coding Phase (1/1) Charac. Sub-Char.		Progr	ram Name: ram # : rammer :	Date:			
		Indicators to measure			Value	Freq.	Resp.
F u n	Suitability and Accuracy	FSu-1a	Program specification change ratio	NB of change to program due to mis-understanding of the requirements  NB Total program		after reviews and at each change request	QA Team
t i o n		FSu-1b	Program specification change ratio	Nb of change to program due to change in environments(legal, organizational, technological)  Nb Total program		after reviews and at each change request	QA Team
a l i t y		FSu-1c	Program specification change ratio	Nb of change to program due to change in the requirements Nb Total program		after reviews and at each change request	QA Team
					l		

Figure C.2.1c An example of Metrics Form Sheet

# Step #5,6,7: Software Testing

Step # 5,6,7,8 measuring the software quality using external metrics.

			LITY ASSURA			):2			
T	nit esting /1)	Progr	'am # :		DATE :				
Charac.		Indicators to measure			Value	Freq.	Resp.		
F n c	Suitability	FSu-1d	Functional Cosmetic change ratio	Nb of change to functions due to Cosmetic change  Nb Total Function		after reviews and at each change request	QA Team		
t i o n	Interoper ability	FSu-6	Matched data format ratio	Nb of transactions with each of the other programs  Nb Total of transactions		Once	Proguer		
l i t									

Figure C.2.1d An example of Metrics Form Sheet

		QUALITY ASSURANCE FORM No					0:3	
Integrated Testing (1/2)		Func	tion Name:		ATE:			
		Func	tion # : yst resp. :					
C	harac. Sub-Charac	Indic	ators to measur	Value	Freq.	Resp.		
F u n	Suitability	FSu-1a	Functional specification change ratio	Nb of changes to functions due to mis-understanding of the requirements		at each change request	QA Team	
c t i		FSu-1b	Functional specification change ratio	Nb Total Function  Nb of changes to functions du to change in environments(legal, organizational, technological)		at each change request	QA Team	
o n a l		FSu-1c	Functional specification change ratio	Nb Total Function  Nb of changes to functions du to change in the requirements  Nb Total Function		at each change request	QA Team	
i t		FSu-1d	Functional Cosmetic change ratio	Nb of change to functions due to Cosmetic change  Nb Total Function		after reviews and at each change request	QA Team	
y		FSu-1e	Ratio DataBase Change	Nb of change request  Total of Data Element		after reviews and at each change request	QA Team	
	Accuracy	FAc-1	Anomalies ratio 1	Nb of anomalies that hinders the function to give results  Nb of tests for the function	144	Daily	Analyst (Dev. Team)	
		FAc-2	Anomalies ratio 2	Nb of anomalies where function give results but wrong  Nb of tests for the function		Daily	Analyst (Dev. Team)	
		FAc-4	Anomalies ratio 3	Nb of good but unexpected results because of script incorrectness  Volume of tests		Daily	Analyst (Dev. Team)	
	Interoper ability	FIn-1	Matched data format ratio	Nb of matched transactions with each function  Nb Total of transactions matched		Once	Analyst (Dev. Team)	

Figure C.2.1e An example of Metrics Form Sheet

## C.3 Measuring Quality

At each step, quality is measured using metrics for the Software. Multiple measurement could be done according to the step you are measure

Table C.3.1 An example of table representing measuring quality and step you are measure

Characteristic	Subcharacteristic	Software Architectural Design	Software detailed Design	Software Coding and Testing	1+1+)4+44	System Qualifica -tion Testing	Software Acceptance Suport
Functionality	Suitability						
	Accuracy						
	Interoperability						
	Security			7704 11111			
Reliability	Maturity (hardware/software/data)						
	Fault tolerance						
	Recoverability (data, process, technology)						
	Redundancy						
Usability	Understandability						
	Learnability						
	Operability						
	Attractiveness						
Efficiency	Time behavior						
There is a second of the secon	Resource utilisation						
Maintainability	Analyzability				·		
***************************************	Changeability						
	Stability						
	Testability						
Portability	Adaptability						
	Installability						
	Replaceability						
	Co-existence						