The DIN/ISO definition and a measurement procedure of software efficiency

Dr. W. Dirlewanger

Kassel University, Prof. i. R.
Dept. of Mathematics and Computer Science
Mozartstrasse 1034246 Vellmar
performance-we@t-online.

Summary and foreword: People like to assign the attribute speed (i. e. performance) not only to hardware but also to software. But software does not have an attribute speed. Software defines - for each user initiated task - a sequence of information processing steps (machine instructions) to be performed. A suitable term for the aspect under construction can only be found when having a second imlementation (Imp2) of the application on the same hardware. We have to perform two performance measurements of the data processing system. Measurement 1, using the first implementation (Imp1), yields the performance value P1. Measurement 2, using the second implementation (Imp1), yields the performance value P2. To find a term describing the property of the software we compare P1 to P2. This yields a measure of „how good" Imp2 transforms the task submitted into a sequence of machine instructions compared to Imp1: Imp2 is less or more efficient than Imp1.

This procedure requires a fitting definition of performance and places great demands to the measurement method. It has to simulate in detail the properties of the user entity and has to deliver detailed performance values. This is done by the ISO 14656 method. It measures the timely throughput, the mean execution times and the total throughput, each as a vector of m values, where m is the total number of task types. Based on this data the following software efficiency values are defined: The timely throughput efficiency I_{TI}, the mean execution time efficiency I_{ME} and the total throughput efficiency I_{TH}, each being a vector of m values. Additionally there is the quotient of maximal timely served users I_{MAXUSR} which is a scalar.

In the talk the ISO method will be presented and explained and it will be shown how the software efficiency values are to be computed. All is supported by examples of real world measurement projects.

Though published years ago the ISO method is little known. Published in the golden age of Mainframe-MIPS may be it was far ahead of its time. But the author is still on fascinated of the abilities of this method. And he is convinced that it may be helpful to save it from falling in oblivion. It is one of the most useful tools for computer performance measuring and Software efficiency measuring. It is applicable also to simulated software for instance in the first phases of a software project. It is mandatory in every software project to specify the enduser requirements concerning the timeliness of the service. These requirements have to be described in the system specification . The ISO workload model is an excellent template for this part of the specification.

A historic aspect is: DIN 66273 was published in 1991 [1]. The software efficiency aspect was not yet part of this National Standard. The use of the DIN performance measurement method for getting software efficiency values was firstly proposed by the author in [3]. ISO took over the idea 1999 in ISO 14756 [2]. It is comprehensively described in [4].

Proc. SOSP 2014, Nov. 26–28, 2014, Stuttgart, Germany

1 The international Standard ISO/IEC 14756

-- Defines the performance term for system performance.
– It describes a method for defining workloads (ISO workload data model).
– It defines a method for measuring data processing performance of a defined IP system using the ISO workload data model.
– It defines the principles of the user emulator.
– It defines a method for rating the measured performance values.
.-- It defines a method for estimating the (runtime-) efficiency of (application- or system-) software which is part of the measured data processing system.

2 Special qualities of the ISO 14756 method

(1) Arbitrary system under test (SUT):
The users in an ISO type measurement are emulated by a remote terminal emulator (RTE) whose input information is an ISO workload parameter set (WPS). The RTE is connected via the real user interface to the SUT. This ensures that any SUT can be measured, independently of manufacturer, configuration, internal construction, operating system, CPU type, system architecture and so on. It may be for instance a mono-processor computer, a cluster of computers or a cloud service. So, even a PC can be compared to a mainframe or to a super computer.

(2) Independence of RTEs manufacturer:
The measurement and its results are independent of the manufacturer or programmer of the RTE and of the computer type used for the RTE. The only condition is that the RTE fulfills the specification of an ISO RTE and naturally can be connected to the SUTs user interface. Measurement results are not influenced by the RTEs individual technical details.

(3) Control of the correct work:
The ISO RTE specification includes a set of control mechanisms which insure the correct working of the RTE during a measurement. If – for instance – the hardware used in the RTE is not fast enough to emulate the planed number of users correctly this will become evident. The same applies to the RTE if it doesn't realize the user behavior correctly and so on.

(4) Nearly every benchmark can be represented in ISO form:
The ISO workload model is very universal. Nearly every benchmark can be brought to the ISO representation. Beginning with the classic “Debit Credit OLTP Benchmark”, nearly any multi-user-multi-programing benchmark, naturally the classic SPEC batch-benchmarks, or benchmarks for today's SAP/R3-applications can be represented as an ISO type workload.

(5) Component tests can also be rewritten to ISO form:
Even computer component tests, for instant speed measurement of a hard disk, can be realized by an ISO type workload; in this case the hard disk is the SUT and the hardware beyond the disk interface (for instance SATA) is the RTE.

(6) Emulated users can be human beings or machines:
Naturally it doesn't matter if the users emulated by the RTE are human beings or machines. For instance the SUT is the central unit of a computer and the regarded interface is the connection to the so-called frontend or a terminal multiplexer. This can also be described by an ISO type RTE.

(7) Forgery-proof by random task stream:
The ISO RTE generates the the task stream transmitted to the SUT randomly. Two subsequent measurements must not have identical job or task sequences as input. Nevertheless predefined global parameters - as for instance the relative task frequencies or think time mean values of the users - are realized according to the statistic accuracy defined in the workload. The ISO RTE works in detail randomly but global deterministic. Intentionally the ISO RTE doesn't use recorded job streams or task streams as a workload. In that case the SUT could be optimized on such a task stream and the measuring results would be falsified.

(8) Reproducibility of measurement results:
ISO results are repeatable. Whenever and whoever repeats a measurement the results are the same. In fact, this is an important difference to common “performance measurements”, that often lack on repeatability

(9) High precision:
The ISO method is a high precision measure method which delivers very exact results.

3 Measurement by emulated users

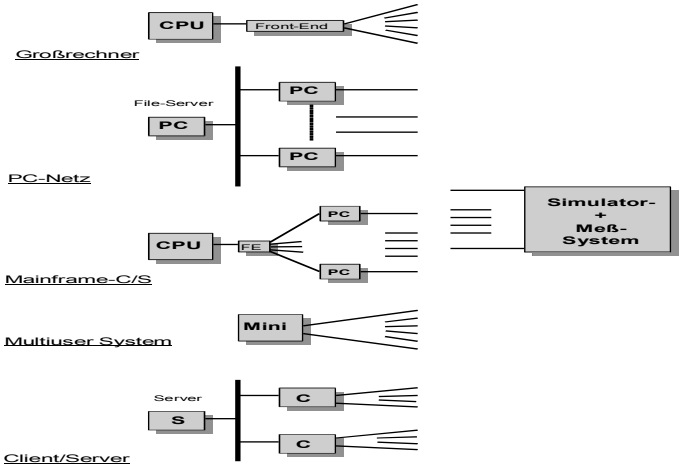

Fig. 1 SUT: Arbitrary type and architecture

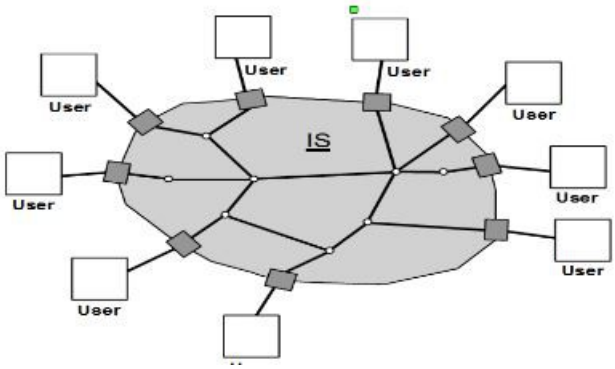

Fig.2 A SUT in real operation

For performing a measurement a RTE is used. It is table driven by the so-called workload parameter set. The ISO workload describes the behavior of all users of the information system. It consists of:

1. Workload parameter set
2. Application programs
3. Operating system command procedures
4. Computational results
5. User data stored in the SUT
6. Parameters for controlling the correct work of the RTE

Steps of the measurement:

1. Install applications in the SUT
2. Load workload parameter set into RTE
3. Run and record; store computational results
4. Testing correctness
   (RTE: correct work and statistical significance of random variables;
   SUT: correct and complete computational results)
5. Analysis of recorded data and computation of performance values and rating values

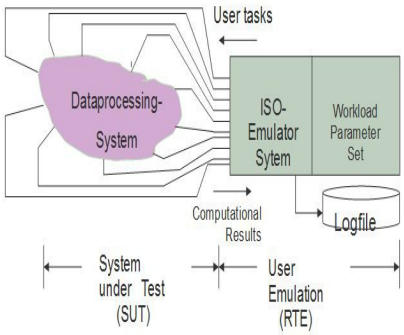

Fig. 3 Measurement Configuration

4 The workload parameter set

Basic parameters:
- Number n of user types
- Number of users of each type $N_{user}(1),..., N_{user}(n)$
- Number w of activity types
- Number p of timeliness functions
- Number m of task types
- Number u of task chain types

Definitions of the w activity types, each described by :
- Input string or mouse pointer action, rules for input variation if there is so.

Definitions of the m task types, each defined by:
- Activity type and wait mode WAIT/NOWAIT for the result of the actual task and
- timeliness functions for completing the task (Example: Maximal 80% within 2 sec, maximal 15% within 6 sec, maximal 5% within 6 sec, none longer than 15 sec.)

Definitions of the u task chain types:
- Chain length (number of tasks) and task sequence

Statistic parameters of the (random to be created) think times of the users
- Firstly: matrix of n x m think time mean values
  (Remark: Think time is task preparation time.)
- Secondly: matrix of n x m think time standard deviation values

5 The measurement run

RTE on
Stabilization phase
Rating interval
Supplementary run
RTE off

6 The ISO performance P computed from the recorded logfile

P is the following triple of vectors: $P = (B, T_{ME}, E)$

where $B = (B(1), …., B(m))$ is the (total) throughput vector.

B(j) is the mean number of tasks of the j-th task type sent from the RTE to the SUT per time unit.

$T_{ME} = (T_{ME}(1),..., T_{ME}(m))$ is the execution time vector .

$T_{ME}(j)$ is the mean execution time of tasks of the j-th task type.

$E = (E(1), …., E(m))$ is the (timely) throughput vector.

E(j) is the mean number of tasks of the j-th task type which were timely executed by the SUT per time unit.

7 Rating of the measured performance values

Throughput rating

$R_{TH} = (R_{TH}(1),..., R_{TH}(m))$ is the throughput rating vector
with
$R_{TH}(j) = B(j) / B_{Ref}(j)$
$B_{Ref}(j)$ is the throughput of the j-th task type of the so called theoretical reference machine.

Mean execution time rating	$R_{ME} = (R_{ME}(1),..., R_{ME}(m))$ is the execution time rating vector with $R_{ME}(j) = T_{Ref}(j) / T_{ME}(j)$ $T_{Ref}(j)$ is the mean execution time of tasks of the j-th task type. of the so called theoretical reference machine.
Timely throughput rating	$R_{TI} = (R_{TI}(1),..., R_{TI}(m))$ is the timely throughput rating vector with $R_{TI}(j) = E(j) / B(j)$

The "theoretical reference machine" is a fictional SUT. It is a not really existing SUT which executes all tasks so fast that all timeliness functions are just fulfilled but none faster. The $T_{Ref}(j)$ values and $B_{Ref}(j)$ values can be computed directly from the data of the workload parameter set. No measurement is needed.

Overall rating: Only if all of the 3 x m rating values are not less 1 the SUT satisfies the timeliness requirements of the user entity. Elsewhere the system has to be rejected due to insufficient response times.

8 Measurement example 1

The diagrams below (see Fig 4 and 5) show the measured performance and the rating of two mainframe computers having the same hardware. Workload: "ISO CC1-I-REP=1" (somewhat modified). The application software is identical. The operating systems are different.

Terms in the diagrams:

Leistungsgrößen –---------- Performance terms
Belastung ----------– Throughput B
Aufträge/sec ---------- Tasks per sec
Auftragsdurchlaufzeit --------- Execution time
AAx ---------- Task type x
Anzahl der EAG ---------- Number of emulated users
Bewertungsgrößen ---------- Performance rating terms

$t_{AN}(j)$ - Mean execution time $T_{ME}(j)$ of the j-th task type
L1(j) - Throughput rating value $R_{TH}(j)$ of the j-th task type
L2(j) - Mean execution time rating value $R_{ME}(j)$ of the j-th task type
L3(J) - Timely throughput rating value of the j-th task type

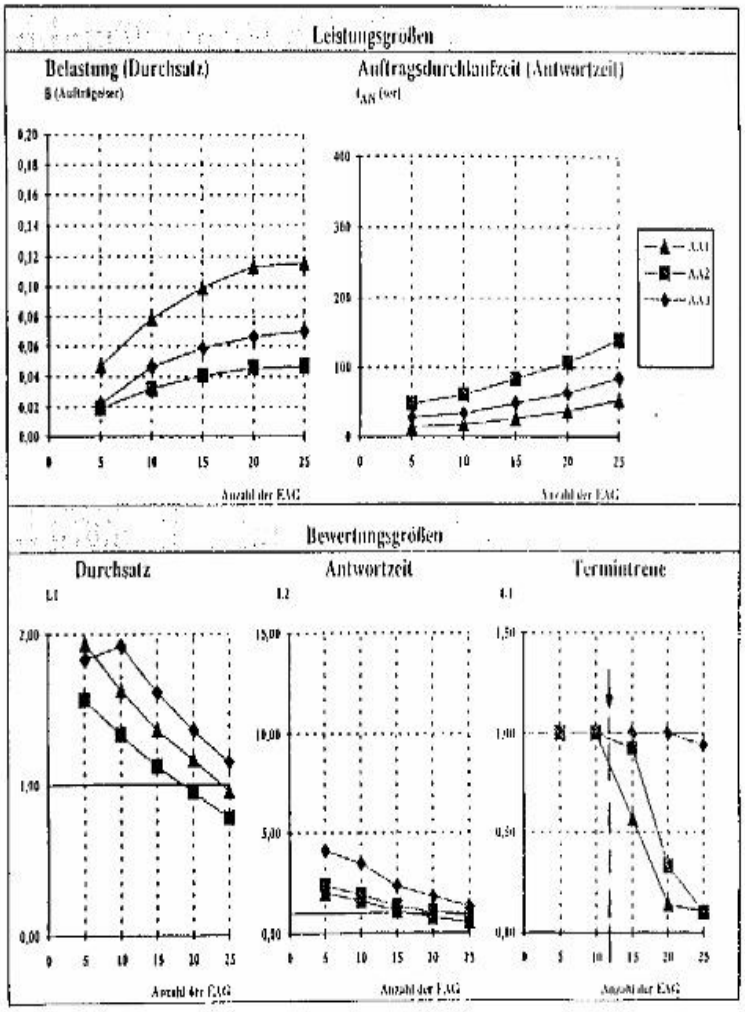

Fig. 4 Mainframe-Nr. 1, Hardware: 8 MIPS
Operating system from "Manufacturer 1"

For more than 12 users is the system no longer timely:
$N_{max} = 12$

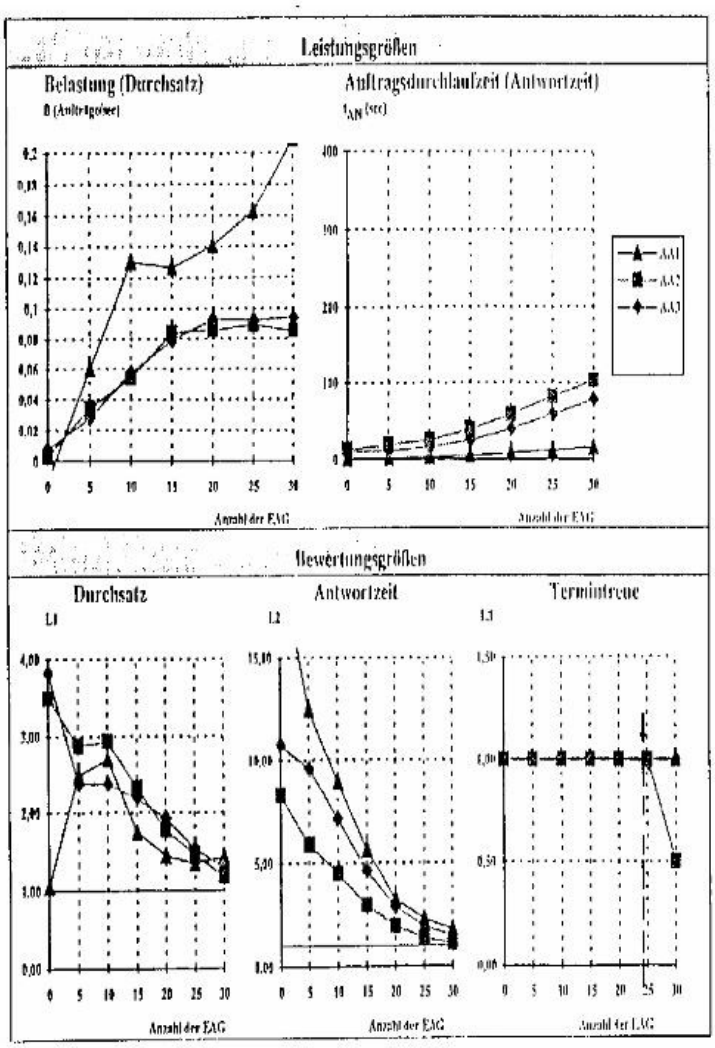

Fig. 5 Mainframe-Nr. 2
Hardware: Same as in Fig. 4 and also 8 MIPS
Operating system: Changed to those of"Manufacturer 2"

For more than 24 users is the system no longer timely:
$N_{max} = 24$

9 Software efficiency in example 1

SW efficiency has several qualities, for instance

- storage usage
- changeability
- maintainability
- runtime

The focus of ISO 14756 is runtime efficiency. Wanted here is the operation systems software efficiency.

The ISO standard 14756 describes the necessity of the reference environment (see Fig.6) in detail. It also describes how to define it for measuring software efficiency. But it does not explain detailed how to compare the measured performance values P1 and P2. To the author's knowledge the ISO working group – which developed the standard 14756 - took it (at that time, i. e. when the Standard was written) for granted how to compare two performance values P1 and P2.

The following comparison of P1 and P2 is done according to a proposal explained in [3] and [4] in detail.

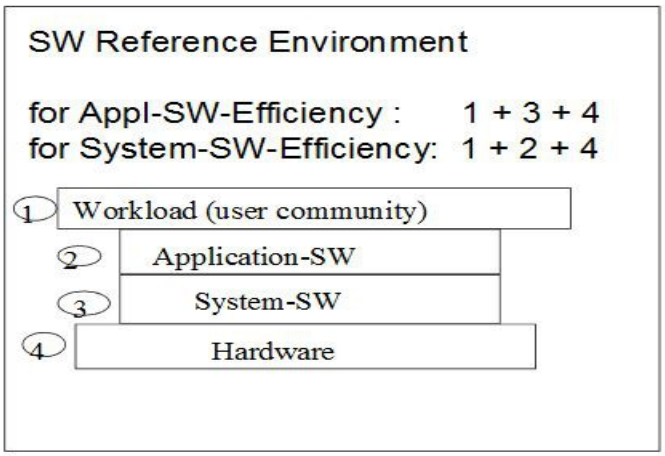

Fig. 6 The reference environment (simplified)

The example: We look for the software efficiency of the two measured systems above(see Fig. 4 and Fig 5).:

The environment is " 1 + 2 + 4" .

The measured terms are:

System 1: $P1 = (B1, T_{ME}1, E1)$
P1 represents 3xm values:

$B1 = (B1(1), \ldots\ldots\ldots\ldots, B1(3))$
$T_{ME}1 = (T_{ME}1 (1)),\ldots\ldots, T_{ME}1(3))$
$E1 = (E1(1), \ldots\ldots\ldots\ldots, E1(3))$

System 2: $P2 = (B2, T_{ME}2, E2)$
P2 represents 3xm values:

$B2 = (B2(1), \ldots\ldots\ldots\ldots, B2(3))$
$T_{ME}2 = (T_{ME}2 (1)),\ldots\ldots, T_{ME}2(3))$
$E2 = (E2(1), \ldots\ldots\ldots\ldots, E2(3))$

The software efficiency values computed from the measured values are:

Throughput efficiency values		(for n=10 users)	(for n=15 users)
Task type 1	$I_{TH}(1)= B2(1) / B1(1)$	0.130 / 0.080 = 1.63	0.125 / 0.100= 1.25
Task type 2	$I_{TH}(2)= B2(2) / B1(2)$	0.058 / 0.035 = 1,66	0.080 / 0.410 = 1.95
Task type 3	$I_{TH}(3)= B2(3) / B1(3)$	0.058 / 0.450 = 1,29	0.081 / 0.060 = 1,35

Mean execution time efficiency values		(for n= 10 users)	(for n=15 users)
Task type 1	$I_{ME}(1)= T_{ME}1(1) / T_{ME}2(1)$	14.5 / 2.10 = 6.90	20 / 4.5 = 4.44
Task type 2	$I_{ME}(2)= T_{ME}1(2) / T_{ME}2(2)$	61.0 / 21.9 = 2.78	80 / 20.5 =3.90
Task type 3	$I_{ME}(3)= T_{ME}1(3) / T_{ME}2(3)$	39.0 / 13.5 = 2.89	50 / 26 = 1.87

Timely efficiency values		(for n= 10 users)	(for n=15 users)
Task type 1	$I_{TI}(1)= E2(1) / E1(1)$	0.130 / 0.080 = 1.51	0.125 / 0.060 = 2.08
Task type 2	$I_{TI}(2)= E2(1) / E1(2)$	0.058 / 0.035 = 1.66	0.080 / 0.038 = 2.34
Task type 3	$I_{TI}(3)= E2(3) / E1(3)$	0.058 / 0.045 = 1.23	0.080 / 0.060 = 1.43

We see: For n =10 users are the throughput oriented efficiency values around 1.5 and the response time oriented efficiency values nearly at 3. For n =15 users are the throughput oriented efficiency values around 2 and the response time oriented efficiency values nearly at 4.

Remarks: 1) Be aware that n=10 users is not the same system software environment as n=15 users !
2) Due to the fact that the lowest N_{max} value of the compared two systems is 12 it is really meaningless to investigate the software efficiency for more than 12 users.

There is an additional software efficiency measure:

$$I_{maxuser} = N_{max}2 / N_{max}1$$

Its value is $24 / 12 = 2$.

Final result:
The operating system of manufacturer 2 is about two times more efficient that of manufacturer 1 when driving a computer center operation as represented by the ISO workload CC1. I. e.: When using the operating system 2 the computer system serves about 100% more users timely.

That was very surprising and was opposite to the meaning of the system programmers of both companies. They were up to this measurement convinced that there would be nearly no difference in the software efficiency of these two mainframe operating systems.

10 Example 2 and software efficiency

The hardware and operating systems are the same as in example 1. But the application software is SAP/R2 instead of the software contained in the (modified) ISO workload CC1 of example 1.

Used were 4 parts of the SAP software:

RF (Finances)	: 40% of the users
RM-MAT (Materials management)	: 30% of the users
RM-PPS (Production)	: 10% of the users
RV (Sales)	: 20% of the users

An ISO type workload was developed. The workload parameter set (shortened):

4 user types
110 activity types
110 task types
21 chain types
3 timeliness functions (mean values 3, 6 and 36 seconds)

Measured N_{max} values:

Operating system 1: $N_{max}1 = 110$
Operating system 2: $N_{max}2 = 170$

Software efficiency

$$I_{maxuser} = N_{max}2 / N_{max}1 = 170 / 110 = 1.55 \quad .$$

Using the operating system 2 instead of operating system 1 the computer system serves (about) 55% more users timely.

11 ISO 14756 and simulated software

The ISO method is not only applicable to real computer systems and its software. It is also applicable for instance to simulated software. Therefore its use is recommended from the first steps of a software project up to the running software in a real computer center operation, and further on in the maintenance phase, for instance when repairing defects or when developing new releases. This is an opportunity to realize high software efficiency (as one of the basic criteria of software quality) along the total life time of the software from its birth to its death. And – even – it can be a help in the first steps of an eventually following project.

A special point of interest is saving man power. Nowadays the practice is to write ad hoc (and highly uncoordinated) new individual test systems for software efficiency whenever the question comes up. Opposite to this it is recommended to integrate this question from the beginning of a software project. This can be done by specifying the workload as an ISO workload which has to become central point of the project specification from the beginning of the project. The workload has to be designed only once and an ISO RTE is to buy or to write only once. The pair workload-RTE can be used by all members and in all phases of the software project. Only and only than when the specification of the software project changes the ISO type workload has to be adapted.

12 Final remarks

The ISO Standard 14756 defines a very universal method for measuring computer systems performance and software (runtime-) efficiency.

A RTE which is implemented according to this standard represents a universal high precision measurement tool and it delivers reproducible measurement results (compare chapter 2).

Implementations of an ISO RTE were realized for instance by the Siemens Nixdorf Computer Company, Telekom and others. To the authors knowledge these companies didn't offer their tools in the market for sale. Free demo versions of an ISO RTE are found on the CD which is part of [4]. An ISO 14756 conform RTE offered in the market is for instance "S_aturn" (see [5]).

The ISO method is not only applicable to real computer systems and its software. It is also applicable for instance to simulated software. Therefore its use is recommended from the first steps of a software project. It is a chance to save man power and to improve the software efficiency of the delivered software product. The ISO workload principle is an outstanding template to describe the endusers demands on his computer center and the installed software.

References

[1] DIN 66273-Serie, *Messung und Bewertung der Leistung von DV-Systemen,* Deutsches Institut für Normung, 10772 Berlin, 1991

[2] ISO/IEC, *International Standard 14756*, *Information technology – Measurement and rating of performance of computer-based software systems*, ISO Copyright Office, CH 1211 Geneve 20, Casa Portale 56. 1999

[3] W. Dirlewanger, *Messung und Bewertung der DV-Leistung auf Basis der Norm DIN 66273*, Hüthig-Verlag, ISBN: 3-7785-2147-0, 1994

[4] W. Dirlewanger", *Measurement and Rating of Computer Systems Performance and of Software Efficiency. An Introduction to the ISO/IEC 14756 Method and a Guide to its Application*, ISBN-10: 3-89958-233-0, ISBN-13: 078-3-89958-239-8, Kassel University Press, 2007 (free extract: http://www.upress.uni-kassel.de)

[5] http://www.zott.net