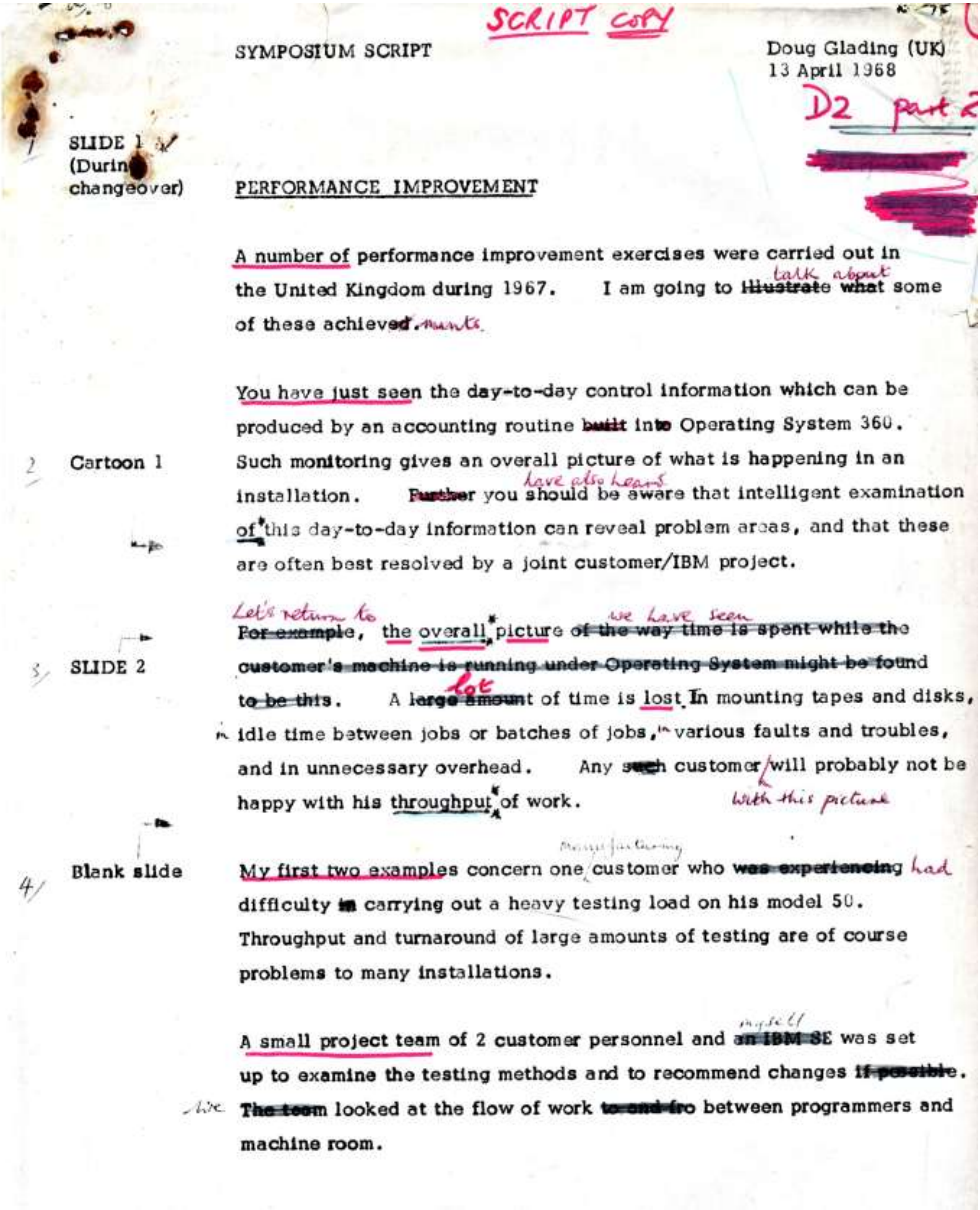
‘PERFORMANCE IMPROVEMENT’ Presentation

1968 IBM European SE Symposium, Cannes -- Script for Doug Glading



SCRIPT COPY

SYMPOSIUM SCRIPT

Doug Glading (UK)
13 April 1968

D2 part 2

[redacted]

SLIDE 1 ✓
(During changeover)

PERFORMANCE IMPROVEMENT

A number of performance improvement exercises were carried out in the United Kingdom during 1967. I am going to ~~illustrate~~ talk ~~what~~ about some of these achieve~~d~~ments.

You have just seen the day-to-day control information which can be produced by an accounting routine ~~built~~ in~~to~~ Operating System 360.

2 Cartoon 1

Such monitoring gives an overall picture of what is happening in an installation. ~~Further~~ you have also heard should be aware that intelligent examination of*this day-to-day information can reveal problem areas, and that these are often best resolved by a joint customer/IBM project.

3 SLIDE 2

Let's return to ~~For example~~, the overall*picture we have seen ~~of the way time is spent while the customer's machine is running under Operating System might be found to be this~~. A ~~large amount~~ lot of time is lost in mounting tapes and disks, idle time between jobs or batches of jobs, in various faults and troubles, and in unnecessary overhead. Any ~~such~~ customer with this picture will probably not be happy with his throughput*of work.

4/ Blank slide

My first two examples concern one manufacturing customer who ~~was experiencing~~ had difficulty ~~in~~ carrying out a heavy testing load on his model 50. Throughput and turnaround of large amounts of testing are of course problems to many installations.

A small project team of 2 customer personnel and ~~an IBM SE~~ myself was set up to examine the testing methods and to recommend changes ~~if possible~~. We ~~The team~~ looked at the flow of work ~~to and fro~~ between programmers and machine room.

5
SLIDE 3

We studied in greatest detail the actual machine room operations during those periods when testing was scheduled. Use of a stop watch revealed that a ~~considerable~~ large portion of any testing period was taken up by the tape, and disk, mounting re;uired by the series of short test jobs. As with most medium-sized machines, the customer was running only 1 job-stream of tests plus spooling. During the mounting of test tapes the machine was often virtually idle.

6
SLIDE 4

We wanted to quantify time losses due to tape mounting. We wanted to get accurate figures without influencing or antagonizing the operator. So we arranged that programmers used a few extra instructions to record how long their program took to "OPEN" its datasets at each test. This code ~~could have~~ can been made ~~on~~ an installation macro.

Of course this method re uires the use of the Operating System facility to "DEFER" mounting re uests for a volume until the program executes an OPEN macro for a dataset on that volume, ~~rather than have the re uests issued at the start of a job-step~~.

We found that over a period 21% of a test sessions ~~were~~ was being wasted in mounting tapes - that is about 13 minutes in every hour. And incidentally the longest time recorded for the change of 1 tape was 27 minutes.

7
Blank Slide

We then looked at the number of tape reels allocated for testing work. The total was about 1000 tapes for this commercially oriented fairly large model 50. Investigating the contents of these tapes showed that few tapes carried large amounts of data. Most tapes only held very little - an average of about 4000 bytes. Only in the final phase of testing prior to handover for production ~~usually re uires~~ did the tapes have much more data, ok

We had a problem - many testing tapes. Tapes which were expensive to buy, expensive to handle and expensive to mount and remount.

8? SLIDE 5

We decided that the Operating System facility of device independence could help us minimise these problems. Basically, this facility allow us to use sequential data sets on disk rather than on tape with only a change in a control card ~~of~~ or in the systems catalog. We found it possibl to switch the contents of about 1000 test tapes onto approximately one third of a 2316 diskpack, ~~which~~ This was ~~able to be~~ kept permanently mounted on the Model 50. A smaller installation might find that part of a 1316 diskpack would be able to do this function.

9 SLIDE 6

This fairly straight forward move gave ~~quite~~ a lot of benefits. It released uite a number of tapes for use elsewhere. It helped ease the job of the tape librarian and reduced the number of issues and returns handled. It meant far less delays for operators to find the correct reel from a ~~number on a trolley~~ trolley-load of tapes and mount it on a tape deck. Life for the operators appeared to become smoother, and of course more work went through the machine.

In an unchanging environment these savings would have been worth at least £10,000 per year - the ~~price of a card~~ the cost of 4 disk drives, or perhaps even a ~~disk~~ model 20. However my next example, from this same customer, shows that we were also able to improve the ~~testing environment~~. way tests were done

10 BLANK SLIDE

The use of disk resident datasets enabled us to satisfy more easily the basic rule ~~of~~ for achieving maximum machine throughput - to keep it working. For another factor which should be monitored by an accounting system is the number of between-job delays of more than, say, 10 seconds.

Our ~~The~~ customer ~~I am talking about in this example~~ now in fact prints a repor of all such delays every 24 hours and requires explanations from the operators for long delays or for numerous small delays.

How did we keep the machine working? Using disk datasets meant that we had turned a good portion of our workload into NONSETUP work. Moving into the environment of priority scheduling systems removes the need to restrict testing into a fixed band of time in a day or in a shift.

SLIDE 7

Tests could be fed into the machine with a specified priority, where they join the queues of jobs waiting on a spool disk. ~~Thus,~~ Whenever there is a gap between production suites or sets of programs a short test job can easily be allowed to execute with no set problems, an no difficulty in suddenly finding or fetching the required tapes.; — "Non Urgent tests" are for me – they will get done eventually.

BLANK SLIDE

Once again, the success of any improvements can be monitored by the normal accounting routines. These should reveal an increased number of tests performed per day, a much shorter average length of job, and fewer occurrences of delays between jobs.

Those were two examples where a small project team produced a number of very worthwhile improvements. Let us look at another customer situation.

EXAMPLE 3

Another factor which should be shown up in any analysis of accounting information is the number of jobs which end abnormally and, if jobs are suitably coded, the amount of time spent doing re-runs.

A common remark made by operators to anyone investigating ABENDs and reruns is "this program always gives a core dump" or "that tape gave errors last time too". A lot of time is wasted by faults occurring again and again ~~This avoidable reoccurrence of such faults is a gross time-waster~~. The faults may be program errors, hardware troubl incorrect job control cards and so forth which have never been reported or reported to the wrong person and not acted upon.

Many customers do have a man performing a "quality control" function. ~~and often this function has been set up as a continuing post after a project has found that considerable savings could be made.~~ Quality Control saves hours of "fault" time.

A project set up to investigate faults will eliminate many unnecessary ones. But almost certainly it will have to design a proper reporting system, which ensures that suitable information is recorded and that troubles are treated with an urgency related to their costliness.

SLIDE 8

We have found that a proper system ~~does~~ require s that ~~that~~ a lot of data has to be written down by the machine operator, such as the programs running, volumes mounted, indicator lights and so on.
In many cases such as an unexplained wait state or ptogram loop the machine is held up for some time. The solution - equip the operator with a portable tape recorder or better still, an IBM ~~224 Lightweight~~ Dictating Unit ~~and~~ To get the machine back on the air as quickly as possible.
An efficient reporting system should be matched by an efficient recording system.

SLIDE 8A

EXAMPLE 4

SLIDE 9

The accounting analysis performed by one of our Aerospace customers ~~In another one of our customers installations his accounting analysis~~ revealed that he was doing the same job, or same sequence of steps within a job, very frequently. In these circumstances the time taken up by the scheduling functions of our control program can become significant - we have an overhead problem.

SLIDE 10

Examining in more detail the steps involved, we found that a very short and simple monitor program could be used to "Link" to each of the required component programs of the set in turn. Normally we would execute the individual programs PROG 1, PROG 2, PROG 3 as separate steps. Control is passed back to the Operating System scheduling functions between each step. We replaced this by a one-step monitor program. Using this technique to LINK just to 2 programs:- the FORTRAN compiler and then the LINKAGE-EDITOR proved worthwhile. On a Model 50 the saving was found to be 8 seconds.

SLIDE 11

Since this customer was performing, about 2000 such FORTRAN compile/link edit jobs a month he saved approximately 4½ hours per month. That is worth in sterling perhaps £200 per month or say £2½ thousand per year. Please do your own conversion into gold. This technique need not be limited to compile and linkedit steps. More complex sequences in normal production suites can show even greater savings.

SLIDE 12

Even in the worst cases the day-to-day accounting information should show, that at least 50% of running time is spent in executing programs. As Terry Halsey observed one of the analyses produced should show the distribution of job rnnning time within various time ranges.

~~If you are in a~~ With a commercial ~~environment~~ customer you should find a situation such as this - a small handful of long-running jobs, consuming almost as much time as all aothers.

Again it is my opinion that it is the responsibility of the customer to appoint somebody to the task of ensuring that programs are as efficient as possible and that the computer is being utilised as near to capability as possible.

19 March

Long running jobs should be examined carefully by this man, or to prove the need for him by a project. A specific measurement they should take is the CPU Utilisation and core used during a job. A recent Installation Newsletter gave a method of finding at a point in time the core used by a PL/1 program. CPU Utilisation can be measured in a number of ways. One way is to modify ~~from modifying~~ the Task Despatching routines within Operating System. Another is to run a special program in the lowest priority partition. This program measures the time ~~to simply measuring the elapsed time~~ to perform a known number of loops round akknown group of instructions ~~in the lowest priority partition of two and~~ calculatings the un-utilised time in the other partition. or partitions

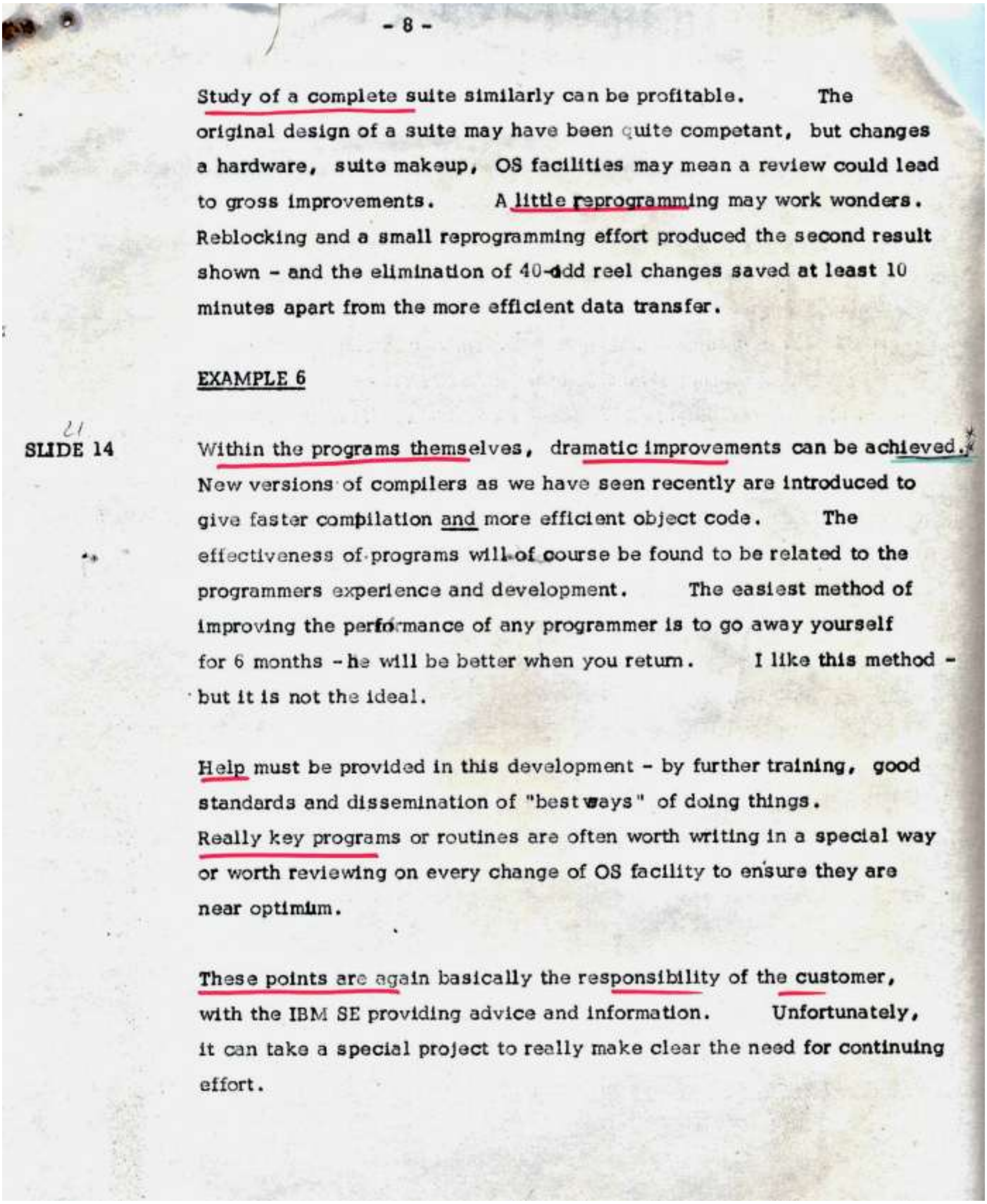
EXAMPLE 5

The last technique was used with ~~For example~~ the Concurrent Peripheral Operations (CPO) program on a Model 40. We found that when performing card to tape (unblocked) ~~utilises~~ 29.8% of the CPU is used. By writing blocked tape of 1600 bytes (a blocking factor of 20) the Utilisation falls to 3.6%.

We should apply this knowledge to any long running job. By using the Operating System facilities of specifying Blocking Factors and Number of Buffers on the control card at execution time we can fill available core, quite possibly reduce the number of tape volumes and make the program more efficient. An example of this - the running time of a 14 minute program was cut to 7.2 minutes.

20 SLIDE 13

Even when multi-jobbing it is still worth doing such studies.
It is wrong to rely on multiprogramming to use up wasted I/O time.
Shorter elapsed job times means units are tied up for less time.



Study of a complete suite similarly can be profitable. The original design of a suite may have been quite competant, but changes a hardware, suite makeup, OS facilities may mean a review could lead to gross improvements. A little reprogramming may work wonders. Reblocking and a small reprogramming effort produced the second result shown - and the elimination of 40-odd reel changes saved at least 10 minutes apart from the more efficient data transfer.

EXAMPLE 6

21
SLIDE 14

Within the programs themselves, dramatic improvements can be achieved.* New versions of compilers as we have seen recently are introduced to give faster compilation and more efficient object code. The effectiveness of programs will of course be found to be related to the programmers experience and development. The easiest method of improving the performance of any programmer is to go away yourself for 6 months - he will be better when you return. I like this method - but it is not the ideal.

Help must be provided in this development - by further training, good standards and dissemination of "best ways" of doing things.
Really key programs or routines are often worth writing in a special way or worth reviewing on every change of OS facility to ensure they are near optimum.

These points are again basically the responsibility of the customer, with the IBM SE providing advice and information. Unfortunately, it can take a special project to really make clear the need for continuing effort.

Two startling examples in 1967 from 2 different customers based on these four points were the reduction of a 32 minute program to 2½ minutes and a 26 minute program to 3 minutes.

SLIDE 15

To review what I have said. I have discussed examples from each of the segments of machine time, where projects following up the indications shown up by normal accounting analysis have given improvements, sometimes dramatic improvements. Performance is gained by reducing volume changing, eliminating between-job delays, controlling faults and cutting overhead time.

23
Cartoon2

I hope you can see where the IBM SE can help the customer by contributing his experience, advice and outside viewpoint to special projects and to the continuing control and review the customer should carry out.

We must aim at creating a more efficient installation with much more of the time being spent in executing good programs efficiently.

Even when this situation is achieved, it will still pay to study the system. For further improvements may still be possible by changing our configuration. In addition, we must consider the other dimension to our picture - that of growth. This is the topic which ~~my colleague~~ Tony Cleaver will now discuss.

DGG/jf
17.4.68

end