the Availability Digest

Worsing on Worsening

February 2009

Ever since the first removal of a moth from the relay contacts of the Mark II computer in 1947, bugs have plagued computing. The search for high availability in computing systems is markedly historic. Over four decades ago, in a 1967 scolding given by Dr. R. A. Worsing, director of Boeing's computer center, to IBM Field Service management, he forcibly attacked what was then the current state of system availability.

Although at the time he would have been happy with a downtime of 2 hours per day (one 9 of availability), many of his observations hold today. We have improved availability in our industry-standard servers by a factor of one hundred (one 9 to three 9s). However, processor speeds have increased by a factor of thousands. One telling comment of his:

> " I'm still uneasily suspicious that, to the manufacturers, a better computer is a faster CPU."

The history of availability improvement lends credence to that statement today.

Dr. Worsing's speech is fascinating and entertaining reading for anyone involved in availability and system support. In fact, legend has it that for many years UNIVAC required all product development managers to read this speech yearly and to sign an annual declaration that they had read it.

SPEECH TO IBM FIELD ENGINEERING BRANCH MANAGERS[1] July 31, 1967

Dr. R. A. Worsing,
Director
Systems Administration and Computing Department
THE BOEING COMPANY

My first reaction on receiving an invitation to speak to you this afternoon was one of sheer incredulity. The few of you who know me well do not, I suspect, regard me as one of your more congenial acquaintances. "Haven't they had enough of me?" I asked myself. "They must be gluttons for punishment."

But being an eternal optimist, I came around to the point of view that at last my preaching, cajoling, threatening and pleading had begun to bear fruit. It will be the first time; but if I can get my message across, it might not be the last.

[1] Thanks to Colin Butcher of XDelta Limited for pointing me to Dr. Worsing's speech. His complete speech can be found at http://www.cs.brandeis.edu/~dkw/Worsing.txt.

So I am acting upon the assumption that you are not expecting any bouquets this afternoon. This is serious business. The problem, so eloquently portrayed by the movie you have just seen, is bigger than both of us. Boeing is not a university, it is not an insurance company. Boeing, because of its size and the urgency of its business, has no trivial problems. It is a problem amplifier, if you like. Whenever anything goes wrong, it goes wrong badly.

Now the area of computing that my people and I are most remote from is precisely the area under discussion today. I have a lot of programmers working for me, and they think they understand software. My hardware people understand logical design and circuitry. But we have no one conversant with the problem of maintenance. Consequently, this is the area we least understand; and when you don't understand, you become frustrated. I say this to give you some understanding of the problem of being the manager of the computing department in one of the world's largest and most rapidly growing users of computers.

The material that I draw upon for my purposes today has been gleaned over ten years as a victim of outrageous hardware. I've run the gamut from part user of a 701 to manager of a $25 million-a-year installation. Today, I have at least one of almost everything that IBM has ever invented. I suppose God has chosen to punish me here on Earth. For some reason, He just can't wait. The consequent baptism by fire puts me in an unequalled position to address you today on the subject of Customer Engineering.

On Customer Engineers

Carl suggested five subtitles to which I might address myself that he felt would be of significant interest to you. I am following his advice and will therefore begin by attempting to answer the question: What do I expect of customer engineers? Well, there's a one-word answer to that - perfection. I expect the machines to work twenty-two hours a day. That's simple enough. Give me a 100% reliable machine, and I will be happy. Allow one bit to reverse itself in the middle of a two-hour run, and I'll be a very unhappy man. It's simple, homespun philosophy, attained by bitter experience.

If this sounds unreasonable to you, consider the following. What would you think of an automobile that, despite a thorough daily overhaul, broke down with 15% probability every day? This is what my grandfather did expect and lived his life accordingly. However, had his generation found this state of affairs tolerable, our generation would have been in the same boat still today. They simply adopted the Worsing dictum: You can't abuse all the people all the time. They finally catch up with you, and you change your ways or perish.

Again, how many of you would board a 707 that, despite two hours of preventive maintenance every day, had a probability amounting to certainty that it would require airborne maintenance at least once between midnight and morning on succeeding Sundays? Or that you would have to board the airplane twice a day every day for a week or a month to be sure of getting from Seattle to New York once?

Let me develop this idea by putting into Boeing's mouth the kinds of repartee that I get subjected to. In this way, you might understand more readily and be more sympathetic to my cause. Now I regard a mainframe, the card readers, printers, tapes, discs, drums, telephone wires, modems and remote I/O units as a single system. If any device is down, the entire system is down. There's no such thing yet as "partly down." Computers are binary, aren't they? This means they are either working perfectly or they aren't working at all. What comfort is it to the programmer to be told that his program almost ran; and had it not been for a loose connection on chassis five, he wouldn't have been here at three in the morning? Now let's change roles.

"Take courage, Widow Robinson, it was only the engines that conked out. The body and wings were in perfect shape. Anyway, we don't make engines; so it wasn't really our fault."

"Members of the Inquiry Board, the plane behaved perfectly all week. Just a loose connection with the main fuel tank in the middle of the Indian Ocean for three minutes. It had passed all the pre-flight diagnostics."

"Of course, we don't guarantee a thing. If you want to be absolutely certain of getting from New York to London, you'd better send three together and transfer the passengers in mid-flight as necessary."

"Ladies and gentlemen, we request your patience at this time. It should only have taken twenty minutes to change wings. However, owing to an absolutely unthinkable and unforeseen problem, we can't get the new ones on properly. If you would kindly disembark, you will be taken to the passenger lounge, where complimentary instant coffee will be served."

"Ladies and gentlemen, owing to an utterly mystifying series of events, our stock of spare hub caps has been found to be exhausted. However, a spare is being flown in from Karachi on Thursday. We do want you to know that this sort of thing doesn't usually happen because we maintain stock levels that are based on years of experience. We know you find this comforting and hope you will find your enforced sojourn in Thule to be an unexpected pleasure."

Well, why is this so ridiculous? Why do you expect so much more reliability with an airplane than with a computer? They aren't any more expensive. They are much more difficult to pilot. They are not kept in a special environment.

I say that the only reason is that airplane customers regard degrees of reliability as unthinkable, while computer customers do not.

And the reason why they don't is that they've been knocked on the head so often, they are incapable of clear thought on the matter. Well, my head is made of concrete; and my vision is quite clear. It is just as intolerable to allow the collection of gadgets we call a computer to admit of unreliability as it is to allow the collection of devices we call an airplane to admit of unreliability.

So to repeat myself, I expect from you simply - perfection!

Anticipating the howls of protest, I want to bring out the following point. It is a point that is not common currency and one which you may not recognize or agree with because you are too close to the situation. It is this: The relationship between the field and the plant is not conductive to satisfactory performance. To put it more plainly, you are not being supported properly by your people back at the plant. And while I regard you as partly responsible for this, I really think that the people back at the ranch - Poughkeepsie, Boulder, etc. - are mainly to blame.

As far as I am concerned, you are the people responsible for reliability. I can't afford to take any other point of view. When the machine is down, you are to blame. But, privately, I recognize the fact that you can't make a silk purse out of a sow's ear; and if Headquarters is in the pork business instead of the silkworm business, then you don't stand much of a chance. Privately, I recognize the fact that reliability is a function of good design, manufacturing excellence, quality control, etc. However, you will never get me to admit that on Boeing territory, precisely because I cannot allow myself to get into the position of holding more than one person responsible for reliability or for anything else for that matter.

I have to force the system to work. I refuse to crutch it by going directly to your Engineering or Manufacturing people. Anyway, your company won't let me. So I go to you, and I demand perfection. It is, in turn, your responsibility to see to it that the machinery you service is designed and fabricated properly. It is this step that I find lacking. My experience has indicated to me that there is a very poor relationship between home base and the colonies.

I don't say that Headquarters doesn't get to hear from you. You wrap every defective part in a pre-addressed box and ship it into the wild blue yonder. From a statistical point of view, Headquarters knows what's going on quite accurately. I don't doubt that. What I do doubt are two things, namely that you receive anything back from them in the way of insight, visibility, statistics, etc., and that they receive anything from you in the way of semantics. By the latter, I mean, for example, they receive a defective five-cent console typewriter connector - just like I'm wearing on my tie clip - and they say, "This makes only 943 defective five-cent connectors this week. We're down 7% on last week. Keep up the good work, lads." What they don't receive is the information that has caused:

a. the entire system to be unavailable for four hours at the cost of $2,000;

b. the rerunning of a two-hour job for $1,000;

c. the delay of all scheduled work to the factory that day, causing delays in the factory at the cost of $200,000.

How often have you ever told that to the factory? How much vinegar do you pour into the pipe-line? Statistics, unaccompanied by semantics, are a cold and lifeless thing; and your job is to breathe the breath of life into your reports to awaken the Rip Van Winkles back home. In the meantime, my job is to render every assistance I can by pointing out your shortcomings.

On Reliability and Availability

The second subtitle concerns itself with the importance of system reliability and availability. Obviously, the foregoing was inextricably mixed with this subject because, really, what I expect from Field Engineering is precisely that product reliability and availability. However, there is a bit more to be told.

Firstly, I want to restate the fact that Boeing is committed to computers. There's no way back. As computers became available, and as we learned how to use them, we slowly hitched ourselves to their tremendous power. But we didn't just replace the machines. Rather, we started doing our business in new ways. We began to get used to expecting new kinds of service. New relationships got under way. And all this novelty was characterized by orders of magnitude of difference in volumes of information, timeliness of information, and reliability of information - both on the scientific and the commercial side. We can now do stress analysis using so many points that squads of people would take lifetimes by hand to carry it out, and we can do this overnight.

We process all the paperwork that the factory needs to bring the parts to the right places at the right times, overnight. The orders for the day are out to the troops before the troops come aboard every single day. And the orders are accurate. The size of our final assembly operations absolutely precludes the possibility of doing this by hand. If we were making one product with little change, it wouldn't be such a problem; and everything would be part of a routine cycle. But just about every airplane is different from its neighbor. We don't mass produce, we custom-build.

Now the great coincidence - a coincidence that is largely not understood, even by many of our people - is that computers became available just when they were wanted. You see, you can't do the stress analysis of a 707, 747, or SST, without a computer; and you can't have a customer-oriented assembly line without one either.

Without the computer, airplane development would have come to a halt. Neither can you have a space program or build thermonuclear devices. This, I think, is a fundamental fact of modern life largely overlooked by most people, including, I regret to say, the computer manufacturers themselves. That's why the sales talk still stresses people-replacement and economy of the old thing instead of capability for the new thing.

So you see, there's no going back. We are committed. In order to build airplanes you need people, buildings, tools, materials – and computers. And to weld these attributes into a viable unit, you must plan and manage that plan; and this entails commitment. And to commit a computer is to state, categorically, that it shall function.

I hope you find this a simple message.

Secondly, I want to take up cudgels with your criteria. I want to discuss, for a moment, how you should be measured and how you should be reported. What is availability?

Each week at our IBM-Boeing Review Meeting, and each month at our meeting with Buck Rodgers,[2] we discuss the incidents of downtime, how long devices were down, what went wrong, how you located the trouble, what you did to fix it, and what you are going to do to prevent it recurring. All of this is necessary, useful activity; but it stops a shade short of reality. The quality of your work is displayed in terms of graphs showing "availability" - and I say this in quotes - and as this quantity increases by percent, we all sit back deluded by the appearance of improvement, basking in the glory of our diagnostic prowess. Sometimes the graph of tape unit performance shows 99.9% availability. It sounds like a Russian election; and in a sense it is, as I shall explain.

How come that in a week of superb equipment "availability," it is possible for our applications people to curse your very bones for causing them daily delays in schedule? Are they getting at the wrong guy? It doesn't seem justice, does it? But, the way we report things today, it is only too possible. You see, in a week of "high availability," it isn't the total length of down time that causes the trouble, it is the frequency. If a tape unit goes down, I don't really care how long it is down, provided the others continue to function. If it's down, get it out of the way; and don't bring it back until you can guarantee it. So it can well happen that "availability" is down but Worsing is "happy" - I put that in quotes, too, you understand.

What causes the trouble is the eye blink downtime in the middle of the two-hour run. The eye blinks don't integrate into anything displayable in charts each week, but the consequent reruns can add up to disaster. So the true availability is total time, minus rerun and delay time. That's the way you should really be judged.

There are some complications, of course. You can complain that if you luck out, the eye blinks will occur in two-minute runs so that total rerun and delay time is not the measure. Perhaps frequency is a better measure.

Then again, it is difficult to obtain some of the numbers. Tape parity errors, for instance, to state the richest source of our displeasure. I will be dealing with this problem in detail at the end of my talk.

Whether you can see all the answers or not, at least I hope that you can see the need for reliability and the need, accurately, to portray availability.

[2] F. G. (Buck) Rodgers led the IBM Data Processing Division from October 1967 to October 1970, when he became IBM director and later vice president of marketing. Rodgers joined IBM in 1950 and retired in July 1984.

On Engineering Changes

[Editor: Think software.]

Now I come to the thorny subject of engineering changes. Engineering changes are regarded as necessary for four reasons, namely: capability, error-correction, reliability, and serviceability. I'll take each in turn.

As far as capability is concerned, and I take this basically to mean speed, I am not interested. Don't waste computer time speeding up the circuitry. I ordered the machines according to a schedule of speed specifications, and I'm content with that. The only exception to that general rule is where you have a design mistake, and the machine has timing troubles. Of course, you have to fix that; but that comes under error-correction.

If you want to speed things up significantly, put all the ECs on in the factory, give the machine a new name, and replace the one in-house. Then I'm never down.

Regarding error-corrections, these, of course, have to be made; and if they are few in number, please install them promptly. If they are not few in number, then I am the possessor of a citrus fruit; and I want neither peel nor pip of it. Get it out after you've got its replacement up and running, and don't have the effrontery to ask for any rent.

I think by far the greatest EC activity is in the realm of reliability. I think you can always find ways of making the machine more rugged. When you build a new model, you like to use the most advanced technology available, consistent with cost and risk. And by definition, this means technology untried in the field. And, in turn, this implies a more than marginal propensity for trouble, particularly in the early stages.

It goes without saying, then, that reliability improvements are going to be rife in the first year to any model. We have seen this quite dramatically in the case of the model 75. When it first hit the floor it was, frankly, a totally useless machine. Its central circuitry was just not up to the task required of it. But since the repopulation in March, it has become perhaps the best of all the 360s despite its speed. Again, in the case of the 91, you threw some $24 million worth of parts into the Hudson - that's a pretty expensive form of pollution by the way - for the same reason.

But when should the stream of reliability EC's begin to dry up? I ask this not as a rhetorical question but for understanding. I really don't know, and I think a frank statement of IBM's goals and expectations with respect to the 360 is about due. The big machines have been out for over a year now, and I think IBM should know enough to be able to tell us.

When you design a machine for a new technology, you don't know what the problems are going to be; so you don't know what troubles you are going to have fixing the problems; hence, the fourth type of engineering, change for serviceability. If the only way to change the [printer] hammers is to hang upside down on a rope from a pulley in the ceiling, then you obviously have to install an EC, i.e., you have to strengthen the ceiling.

Serviceability means more diagnostics in the allotted preventive maintenance period and less time needed to locate and repair faults. So let's have serviceability engineering changes.

Having covered the reasons for ECs, I now want to mention the vexed problems of scheduling the time needed to install them. What are the premises? Certainly, if the need for improvement is recognized, the improvement should be incorporated. But if the machine has been working reasonably well since the need was recognized, one more day won't do it any harm.

Parenthetically, if it hasn't been working well, then the EC is not regarded as such but as part of a program to get the machine up and running. Furthermore, it takes time to install changes; and I

don't want to use more time in installation than I save by installation. And I want preventive maintenance time taken entirely for preventive maintenance. I can only allow EC installation provided the standard crew performs its normal procedures without any interference. In addition, ECs, by their very nature, change the machine to something different; and the consequences of this cannot be predicted with complete accuracy. Therefore, I cannot be certain that the machine will recover from the trauma.

This is a pretty formidable set of conditions and restraints, and I have no formula for success. What I think I would prefer to do is give me:

a. A plan of philosophy and expectations for each machine, updated as experience is acquired.

b. A statement of the risks involved in allowing ECs to accumulate.

c. A general plan for acquiring the time to install the main bulk of the ECs, and

d. A coordinated schedule for the installation of ECs requiring lengthy installation or having high probability of causing subsequent morning sickness.

On Preventive Maintenance

That concludes all I have to say concerning engineering changes and leads me to the next topic, that of preventive maintenance.

I understand the need for preventive maintenance and am happy to schedule a period each day for its execution. This is supposed to purchase my ticket to perfection, reliability, availability, and all stations to happiness. What you do with your preventive maintenance time is not regarded as any affair of mine. I put on my walking shoes, and I sneak past squads of engineers armed to the teeth with ring binders and scopes; but I never intrude. And if all were well, I'd stay clear out of the way. But it isn't. How many times does a device that has been running perfectly all week, crop out within the first hour of preventive maintenance on Friday? And not just trivial devices, either. How often has the two-hour preventive maintenance period been followed by a four-hour post-p.m. recovery session?

Now, I'm the first to admit the existence of Heisenberg's [Uncertainty] Principle: You can't measure the temperature of the bath water without changing it, and I suppose you can't test equipment without imposing some strain on it. But sometimes I think you're too brutal. You seem to split the patient's head to find out if he has any brains.

But it seems computers are designed for smooth running, not for diagnosing. Perhaps we need a diagnostic-oriented computer. We already have primitive diagnostic circuitry in some of the 360s. Perhaps this is the first step towards the fail-softness you keep telling me is just around the corner.

But today I get the very definite feeling: a) that the machine hates being tampered with and, b) that your techniques aren't sufficiently scientific or comprehensive.

Why should we have better diagnostic programs than you? But we have. By your own admission, there's no better diagnostic than a production program; but you can have a copy of that program any time you like. You have access to everything we've got, plus your own staff, so really there's no excuse for handing over the machines to us without a guarantee of current perfection, is there?

What is haunting both of us, I think - and here I find myself banded with you against our common enemy, the Poughkeepsie Business Machine Company - is that with today's machines, it is very

difficult to tell sometimes whether the machine is up or not. Tapes may be spinning, messages frantically typing themselves on the first generation console typewriter - they're so cryptic anyway as to be indistinguishable from random sequences, lights flashing, yet nothing but chaos being created. Another way of expressing this, perhaps, is that production work looks pretty chaotic. We're OK when we get a hard stop. All systems are stopped, we fill out a card, hand it to the C.E.'s, and go off for a smoke. But IBM has invented a thing that I hereby name the "soft stop." It's dead but gives all impressions of being alive - like some people I know. It takes very clever operators, assisted by knowledgeable software people often, to issue the death certificate.

What's it going to be like with multiprogramming? Have you considered this problem in the multiprogramming environment?

My mind boggles at the contemplation of the interaction of coexistent programs on a soft machine. Are you equal to the task? Or are you going to be a reincarnation of the Grand Old Duke of York? [3]

Well, to summarize my thoughts on preventive maintenance, I will merely repeat that I'm in favor of it; but I feel that there is still plenty of scope and requirements for improvements in your use of the time.

On The IBM 360

My last subtitle is what I think of the 360. This, of course, is really a speech in its own right. I don't have time to give more than a brief summary here today.

Firstly, I think it is appropriate to state my reasons for choosing the 360 in the first place. We chose it for its peripherals. Peripherals are the key to data processing, not mainframes. I should point out here that I do not regard the 360 as a computer; and for that reason, we do not use it as such. We do our computing, our arithmetic that is, on the machine of another vendor. The 360 has a short word-length. It is a hexadecimal machine that makes it effectively even shorter, and it does not have floating-point round. You can say that the model 91 commits round-off error faster than anything on the face of the earth, while the first lecture in the introductory course of any Numerical Analysis Program states that this is the fundamental problem of computation and the source of all evil.

It is one of the curious mysteries of IBM, the omission of rounding in the 360. It would be less of a mystery if IBM were to come right out and say that the 360 is intended for data processing and not for computing.

However, back to the mainframe theme. Data processing is essentially the business of obtaining correct bits, storing them on something for periods of time, and being able to read them from storage whenever I like. Questions of language and speed are secondary. The question of reliability, as you know by now, is the primary one.

As you also probably know, the most unreliable device in data processing is the conventional half-inch magnetic tape. This has caused The Boeing Company's Commercial Airplane Division more trouble by an order of magnitude than anything else. It suffers from problems of environment (dust particles), electronics (it is highly analog rather than digital), mechanical wear (oxide on the tape), mechanical obstinacy (reels spinning in opposite directions). Half-inch magnetic tape, even today, is a Rube Goldberg device, very little removed from the string-and-pulley device of fifteen years ago. Yet we continue to commit the invaluable information of the company to its rugged surface. You might just as well put to sea in a sieve!

[3] A reference to a children's rhyme about the Duke of York. When his army was halfway up the hill, he didn't know whether they were up or down.

So that was the first problem to solve. We had to get off the conventional half-inch magnetic tape onto some sort of twentieth century device. And if this were all we achieved by going 360, it would still be a significant improvement to the business of designing and building airplanes.

> [Editor: At this point, Dr. Worsing goes into a long diatribe about trying to find a readily-available data storage device to replace magnetic tape and his failed attempts to try to use the new tape cartridges called “hypertape.” He concludes with a reference to Boeing’s early attempt to use IBM’s Data Cell Drive, which could provide a system with three gigabytes of disk storage.]

In time, we will know how to use this rich array of devices. It will no longer be necessary to prescribe leeches for every ailment; instead, we'll let the punishment fit the crime.

We'll know enough about the economics. We hope we'll be able to find out how many to order of a particular device to ensure the availability of one. We hope that our programmers will have mastered the subtleties of Job Control Language. But when the day of revelation dawns, please don't announce System/7.2832, the 9-bit bytes, PL/2, Supertape, Dinky Disk, Data Jail, excommunication devices and voice input - however trivial the conversion. We have enough to keep us busy for the next decade, and all we need from the manufacturer is unprecedented availability. You, the purveyors of availability, are now in the forefront of the struggle to make this vast complex of possibilities a viable, economic reality. I am sure you are equal to the challenge, and I am confident of your continuing ability to ensure its eventual success.

In Conclusion

I thank you deeply for the opportunity of conveying these thoughts, impressions, criticisms, questions and aspirations that I hope have been taken in the spirit given and would be happy to engage in vigorous debate or discussion if there are any questions from your side.