

**PLAYING THE WRONG GAME:
MEASURING PROGRAMMER PRODUCTIVITY
IN A 4GL ENVIRONMENT**

Gene Harmon
AH Computer Services, Inc.
8210 Terrace Drive
El Cerrito Ca. 94530

There are many answers which were developed in response to some extremely difficult questions posed as part of the development of the period we now term the third generation of computer technology. The question considered in this paper will be the one which asks: "How does one measure programmer productivity"...to which we will add the phrase : "in a 4GL environment." This question is being heard with increasing frequency lately, and it, in turn implies a series of other questions which can be thought of as part of an agenda which has been left to us as part of our heritage from the third generation of programmers, their managers and their users. I see these other implied questions as the following:

- Is there such a thing as a programmer in today's 4GL environment?
- What is the definition of work for today's knowledge worker?
And who does what?
- What is the end-product desired in such an environment?
- And. . . what is the best way to ensure a successful delivery of that product?
- What are the impediments to excellence?

These are all questions which come up and are covered with endless regularity at this and every other conference in the world, and my experience is that when I attend these types of lectures, I am both amused and enlightened; and when I look around, (especially if it is a good speaker,) I can see a sea of nodding heads. So, it would be a mistake to say that yet another utterance by me will solve these seemingly perennial problems . . . yet, it does seem that, over the years, the environment provides enough change to prompt for a serious re-evaluation of our tasks in the workplace. Some of the new by-words in the market-place are :

- | | |
|--------------------------------|-----------------------------|
| - Competition | [the Japanese] |
| - Responsiveness to the Market | [development cycle] |
| - Lean and Hungry | [fire middle management] |
| - Innovation | [RISC architecture] |
| - Quality | [Chips, Cameras, and VCR's] |

Although all of the words have an effect on each and every one of us, no matter what our job in the marketplace may be, it is the last word "Quality" which would seem to exert the most leverage in our topic for today. No, this is not going to be a talk on the merits of quality control; so please don't have the reaction of the American who was caught in the wrong place in the jungle with his Japanese and French co-entrepreneurs . . . immediately prior to execution they were allowed to say one last thing: the French person asked to be allowed to sing his national anthem; the Japanese wished to deliver his lecture on quality control one last time; and the American requested that he be shot before the Japanese so that he would not be forced to listen to the lecture one more time. So, the only stave in the quality control lecture I shall use to beat you over the head with is the following quote from David Halberstam's book, "The Reckoning" :

" . . . quality was not some minor function that could be accomplished by having some of the workers at the lowest levels attend a class or two, or by appointing a certain number of inspectors to keep an eye on things. True quality demanded a totality of commitment that began at the very top: if top management was committed to the idea of quality and if executive promotions were tied to quality, then the priority would seep down into the middle and lower levels of management, and thus inevitably to the workers. It could not, as so many American companies seemed to expect, be imposed at the bottom. American companies could not appoint some medium-level executive, usually one whom no division of the company particularly wanted, and for lack of something better to do with him, put him in charge of something called quality. The first thing that an executive like that would do, Deming said, and quite possible the only thing, was to come up slogans and display them on banners. If the company treated quality as a gimmick or an afterthought, then true quality would never result. Above all he was saying, quality had to be central to the purpose of a company."

This book is well worth reading, and one of the most telling incidents in it concerns an American visitor to the Nissan automobile factory who politely asks where the holding area is for cars which did not pass inspection and must be repaired. There, of course, is no need for such an area as the Nissan factory, although each American factory has a repair bay at least the size of a football field to hold its rejects.

Now, the lesson here is not how to make better cars. The point is that anyone who makes a commitment to quality and who positively refuses to compromise or surrender that commitment, can expect superior results. Another lesson this book makes, in a very excellent point, is that there is no one operational improvement that one make which suddenly transforms your world; rather, it is a series of reasoned, methodical changes made by a team striving toward a goal of excellence.

Considering the plight which really does affect us all, I think it

behoooves us to consider the questions posed earlier in the talk, and to do all that we can possibly do to provide answers to them--which just might mean that our jobs ten years from now will still be in the computer business rather than in the fast food business. If this statement seems at all bizarre or laughable to you, as you are enjoying the ambience of this resort atmosphere on tax deductible money, remember that just 15 years ago, people with similar messages were being laughed out of the boardrooms of the big three auto makers who are now existing at the sufferance of voluntary quotas by Japan and Korea, and by the mercy of government bailouts; to paraphrase Sinclair Lewis: "It Can Happen Here!"

Question #1: Is there such a thing as a programmer in today's 4GL environment?

-Old 3GL role Vs. New 4GL role.

I speak today as a refugee from the 3GL environment where I was first a trainee, then a programmer, then a programmer/analyst, then a senior programmer/analyst, then a systems engineer, then a technical consultant, then a contract programmer, and now a senior consultant who is back to typing in his own code (even a programmer got to hand his coding sheets over to a key-punch operator back in the old days). In some ways, it seems I have come full circle while theoretically having advanced to the 4th generation of programming languages. . . while I can enter three statements which will cause a processing screen to be built which can add, update, and delete a record with many fields in it, I find myself pondering escape sequences to send commands to laser printers. What has changed? Primarily; the combination of new hardware and software has released business organizations from the total dependance upon a hierarchical data processing staff. Although there may be large DP staffs associated with 4GL environments, they are typically populated by individuals with a wide range of skills. They are no longer limited in their activities by archaic job titles with corresponding specialized narrow ranges of action. They consider it a part of their normal activities to consult with the user, design databases, write and test programs, and consider new hardware/software options as solutions to user requirements. In a word, the old definition has "vanished".

-Old Specialized Skill Sets Vs. New Omnibus Skill Sets

Repeating my theme of initiation into the world of data processing via the third generation, I can recall taking the IBM programming test which was to measure my aptitude for programming. High emphasis was placed on that test, and I imagine that the sales representatives for that organization were required to take an entirely different test. The litmus test for those who would succeed in today's world would seem to me to be a hard one to construct. The requirements for success are so varied and so demanding that it seems unfair to ask any one person to fulfill them all. It is not unusual nowadays for a two or three person staff to be responsible for the complete data processing requirements for companies doing millions of dollars of business a year. These circumstances require that from the resources and talents of these few people; all user interface, hardware and system require-

ments, software and programming needs, vendor interface, internal management, and planning for the future- must be satisfied. Such high demands cannot be met by the static requirements of job descriptions dredged up from old 3GL references, they require a data processing staff willing to learn a wide range of technical, business, and interpersonal skills; whatever the cost, and wherever they can find them.

-Old Vendor Knowledge Pipeline Vs. New Market Pipeline

The 3GL workplace was one in which information about new products, whether hardware or software, arrived via the vendor salesperson, from some comment a manager might hear at DPMA dinner, or perhaps in the pages of ComputerWorld. Information of this sort might or might not filter down to the person actually dealing with the problems, and users typically had to make do with DP solutions which have been developed five to ten years prior, and hope that his latest requests might be handled by some anonymous maintenance programmer. Much of that scenario has now changed. It is not uncommon to see the entire staff of a 4GL installation in attendance at popular user group meetings, even those which might require traveling hundreds of miles. At these meetings each staff member has the opportunity to sample the wares of dozens of vendors and attend technical talks where he might encounter an entirely new solution to his problems entirely by chance! This new market pipeline of access to vendors and peers represents both a challenge to select from the marketbasket of offerings and a requirement to maintain current knowledge of the latest hardware and software solutions.

Question #2. What is the definition of work for today's knowledge worker, and who does what?

- Re-evaluation of on-site and off-site time
- Redefinition of roles

Much of what has to be dealt with here for this question is a natural outgrowth of the first question; the new responsibilities foster a new definition of tasks for today's knowledge worker. In order to provide the user a satisfactory solution to his problems, the person providing that solution can no longer exist in a cubicle laying out lines of code; the systems analyst can no longer rely on tried and true methods of batch processing to present monthly and annual reports from which the accounting department must extract the information needed to generate financial information. Time well spent by the worker of today consists of such varied tasks as reading the latest technical magazines, experimenting with the latest software package, and attending every users group meeting possible. These activities are no longer the domain of just a privileged few in the DP organization, they are an obligation of all who would maintain functionality in a 4GL environment. It is necessary that all staff members who have the task of implementing user requirements also develop the expertise to extract those requirements from the user; those who have no direct feeling for the user's needs will not be able to focus their full implementation talents on a refinement and enhancement of those needs. There is a book written by Nevil Shute about a religion which has sprung up amongst a fraternity of aircraft engineers in Asia which contains the

lines:

"Aeroplanes come to grief because of wrong cravings and wrong hatreds and illusions in men's hearts. One of you may say, "I have not got the key to the filler of the oil tank. I cannot find it. I looked yesterday and there was plenty of oil. It is probably all right today." So accidents are born, and pain and suffering and grief come to mankind because of the sloth of men. . . . It was the same message that he had preached so often in the hangar at Bahrain, that the maintenance of aeroplanes demanded men of a pure and holy life, men who would turn from the temptations of the flesh to serve their calling first."

The analogies to a DP worker are easy to see here; though one might reasonably draw the line at certain sacrifices hinted at, the spirit of dedication to the task at hand is surely admirable.

Question #3. What is the end-product desired in such an environment?
. . . and what is the best way to ensure a successful delivery of that product?

Questions such as this one surely lurked in few programmers' minds of the third generation; usually any program that ran bug-free to end-of-job was considered a roaring success...if one could do it quickly and often enough it was a passport to the next job-step or to the next job for more money. Data processing empires were built and turfs were staked out in fights over personnel and budgets; although many of these fiefdoms still exist, they are slowly being undercut by users getting their hands on PC's and sometimes using their own budgets for small mini-computers which can be tailored specifically to their needs. Thus, the end-product has really been redefined by the new technology and the ability of the users to insist that their needs be directly satisfied. This new definition of the end-product means that the 4GL DP person must learn how it is that the business is run, and then make sure that the system of programs used accurately reflects that flow; this is quite different from capturing information from after-the-fact forms and then churning out reports. The main difference seems to be that while a 3GL allowed one to report upon the state of the business, the new 4GL allows day-to-day operation and control of all business operations. These new abilities mean that the end-product of DP endeavors is now very closely tied with the goals of the business which it serves. An effort must be made to make sure that these business goals are fully understood, lest the new technology hinder rather than help them.

Question #4. What are the impediments to excellence?

Everyone, I'm quite sure has personal experience with these impediments, whether they be political blockades or short-sighted economies. Many times it is impossible to exert the leverage necessary to overcome these impediments; however, it is possible not to construct them yourself. In many environments, it possible to get users to sign off on systems which fall short of doing the job that could be done,

and then allow yourself to become the victim of inertia and merely maintain your position instead of going forward... do not let this happen. Allow yourself to be open to new methodologies, but do not surrender yourself and all your existing systems to them.

CONCLUSIONS

THE CYCLE OF IMPROVEMENT

- Commitment to Quality
- Statement of Goals
- An Agenda

It seems to me that once you are determined to do the best you can possibly do in your particular environment (or possibly change the environment itself), a commitment to quality which emanates from management throughout your organization is an excellent first step in the cycle of improvement. Remember that each step of this cycle, as well as the total aspect of quality consists of many smaller increments, and of many choices. Remember also that some of these choices may be difficult ones; ones which may not be accepted by everyone, and with which you may not be in love yourself. Lastly, always remember what Walter Bagehot, a 19th century British essayist said:

"One of the greatest pains to human nature is the pain of a new idea"

It is also necessary to state what the goals of your particular organization are . . . note that is easy to develop beautiful systems while, at the same time, failing to meet the needs of the user of those systems. The goals developed must be in alignment with the reason for your organization's existence. Once you have made you commitment to quality and stated your goals, you must then develop an agenda, or a plan for action. You should not fall into the trap of running your organization as the Harvard Business School would run it:

". . . too much information, too many options, too little feeling about the product."

Just say "NO" and try to do a few things well before going on to the next group of a few things that need doing.

My own feeling is that the cycle of improvement is very much an iterative process which very much demands constant re-examination and tinkering. Before it becomes necessary to include the measurement of programmer productivity into this cycle (for the want of something to do with your time) I hope that all of us here have graduated to the 5th GL and are totally occupied with a brand new set of challenges.