CONSUMER PROTECTION
IN THE DATA PROCESSING FIELD

Jean François
Ministry of State for Urban Affairs*
Ottawa, Ontario K1A 0P6

ABSTRACT

The data processing field is a domain where several professional groups of people must work harmoniously together in order to produce satisfying output. The result is that when complex systems are made to work we may have good reasons to believe that it contains a miracle and when they don't we can expect serious problems, some of them apparently undeserved.

The question is what protection do users have when they deal with software or hardware companies? Is there a safe way to do business? This paper tries first to explain why the data processing field is so risky and then convince the users that their best protection are their knowledge and ability to use them. This article itself refers to the live situation of a minicomputer system which did not want to die.

RESUME

Le domaine du traitement des données compte plusieurs types de professionnels qui doivent travailler ensemble de façon harmonique afin de réaliser un extrant satisfaisant. Lorsque tout fonctionne bien, ou est prêt à crier au miracle alors que dans le cas contraire, on peut s'attendre à des problèmes graves, certains apparemment non dûs.

Quelle protection ont les clients lorsqu'ils font affaire avec des compagnies de logiciel et de matériel? Y-a-t-il une façon sûre de faire des affaires? Cette communication essaie d'expliquer pourquoi le domaine du traitement des données est si rempli de risques et de convaincre les usagers que finalement leur meilleure protection réside dans leurs connaissances et leur habilité à utiliser ce matériel. On y donne comme exemple, la situation d'un système utilisant des mini-ordinateurs "et qui ne voulait pas mourir".

---

* The views expressed are those of the author and do not necessarily represent those of the Ministry of State for Urban Affairs.

CONSUMER PROTECTION

Consumers in the data processing field include a certain number of cascaded users who provide systems or services to end users. At some point in time or another, they all have been consumers of data processing products or services. This picture as is, is not different from the construction business or the car industry, however, it appears that the data processing field is an area where one can get burnt or robbed quite easily.

The immediate reaction which comes to our mind is to ask, "Is there not some sort of protection when we purchase a computer system?" The answer is yes, but practically in most cases when a problem arises it is because some important factors or parameters have been left aside sometime during one or several phases of the acquisition of a system. Therefore, per default, the end user will collect and bear the consequences of everybody's mistake ! Undoubtedly, the next reaction is to say "True, but if I am careful, I should be safe". Again, when designing and writing the specifications of complex systems, it is not possible to think of every potential problem simply because they are too numerous and some of them will only appear for example when a system is in operation or when it is more heavily loaded. At this point in time, what was taken for granted suddenly no longer works satisfactorily and this is where most "undeserved" problems may provoke a brain storm.

In order to minimize those imponderable, the behaviour of the data processing community must be taken into account.

BEHAVIOUR OF THE DATA PROCESSING COMMUNITY

The basic three bodies of knowledge : hardware, system software, and application which form the data processing field are interdependent, distinct and have no clear delineation between them. In addition, the hardware and the system software must both be working to run the application programs and in turn the real proof that both the hardware and the system software are working are obtained when application programs are also working satisfactorily.

.../...

From this, it is easy to see that the environment to cultivate the problems is extremely fertile and that in case of hard to find problems, the loop hardware, system, and application, may be quite hard to open.

This status of fact is not totally abnormal when we think that a hardware specialist can spend all his life in the design, reliability, packaging, etc. of electronic circuitry that a software specialist can spend a lifetime in learning and decorticating operating systems and that finally the specialist in applications will never finish learning languages, the use of packages, etc. For him, the machine is a remote black box that he learned to master somewhate similarly as a dowser with his forked piece of wood. It works but he cannot really explain why.

The above look at the three bodies of knowledge which cohabit side by side is not enough to explain why in some cases projects are discarded before they are finished, are never finished, nor debugged and sometimes have an abnormally high turnover of personnel at all levels.

It is well understood and accepted that if the interfaces between various sections of a project are well defined then it is enough to ensure that at least a workable system will emerge some day. This may not always be true because each individual belonging to the data processing community is governed merely by a single motive : the survival. At the lower level individuals are concerned with the survival of their position and at higher level with the survival of their unit or company.

The true means for survival are never written, rarely said, and never repeated. The basics can be learned quickly by observation, however, according to the skills and the nature of some individuals, some strategies may be quite difficult to uncover especially when a valid reason is there to cover up, For example, why a brilliant individual is switched before the end of a project, has not started at the beginning, has never had the time to write proper documentation or why consultants are suddenly brought in.

.../...

At the "doer" level, programs or designs are usually made in a complex way for two reasons,first because people are purposely placed at the maximum of their capabilities, -although some persons may be overqualified in some other area- and secondly because it is one way to make the program undecipherable by somebody else. In addition, skilled "doers" can make a program practically unchangeable almost by adhering too closely to the specifications.

Briefly, the top management wants primarily to have a product in time which meets the specifications and which is appealing to potential customers. The middle management has to be careful that no individual below him will know everything and therefore will be in a good position to supplant him or find mismatches in the design. Finally the "doers" must be careful that they ensure their future in being irreplaceable "without me nothing works". Also most "doers" know that it is not in forging a reputation of an excellent technician that they will progress in the hierarchy.

Putting all the above variables in a project equation we see that it can hardly produce a happy user in the long run. However, an equilibrium is reached whereby the quality of a final product, as an average, is equal to the minimum acceptable by the users. In other words, users as a whole have just what they deserve.

Who is to blame and who suffers ? - probably as a whole the data processing community and more specifically the end users. This is partly due to the competition, our set of values and the nature of the individual. Having seen in which environment a product is finally marketed, there is another aspect which remains to be examined, it is the support which is given to this "average product" once it is sold.

Companies are not all equal in size, product quality, support, type of clientele, etc. However, they are most likely all average. The good point of one may correspond to a bad point of another and this is why they stay in business.

.../...

CONSUMER PROTECTION

A company whose aggregated characteristics fall below average will go broke. How to stay alive?

As a rule you must receive more than what you give. Should you have no reserve, the equality means that you are on the verge of bankruptcy. A company who thrives must have a way to receive much more than what is gives. Two things should be looked at carefully, one is that the quality of the product may not be as good as what it is supposed to be, and two, that the support given to the product may be inexistent or priced separately. We should also remember that miracles have not yet created money. It is nice to know that we are somewhat robbed by a company who will stay in business. Again, who gets robbed? -mainly the end user and the employees. The phenomenon is quite general. It has been proved that during wars proportionately more country boys who believe and obey more readily got killed than the generally more elusive city boys. The latter are better equipped to survive.

The above portrait of the data processing community will help to understand the problems which we have encountered at the Ministry of States for Urban Affairs while we were dealing with computer manufacturers, consultants and hardware maintenance firms during the acquisition and implementation of our minicomputer based system.

STRUGGLING TO GET OUR DUE

Back in late 1973, we thought of implementing in-house, a minicomputer based system to be used in a scientific environment and as a remote job entry station to external service bureaux. With this in mind, we first started to do our homework, that is to say, studying our needs with respect to computer services, then we submitted a recommendation and a proposal.

At the time that the proposal was written, we had a good idea that what we had in mind was feasible,and that at least one manufacturer could submit an acceptable offer in the price range of our budget and with already existing system software. That was a decisive turning point. Any philosophy or abstract terms brought in at this stage were automatically discarded unless we could translate them into concrete and realizable segments.

.../...

This at least up to a point where we knew that later, we would not be faced with colossal differences between reality and expectations.

In order to take only calculated risks and to protect ourselves, we adhered to the following commandments :

1. The contemplated system can be made working in the first stage with relatively few changes.

2. The quality of the job to be done could not be questioned, the emphasis being accuracy first, speed second.

3. The envisaged system had to be expandable and,

4. never let a consultant put you in his claws.

The writing of our specifications.

We had two choices : either write the specifications including all known possible details in a mountain like manner requiring a life time and being not specifically safe, this would cost a small fortune, or ensuring ourselves to include all the essentials, with a detailed hardware configuration. We would then have to do the rest of the work ourselves with some outside help.

The second approach was the only acceptable one as experience proves that it is only when a system is in operation that the real problems and the bottlenecks appear. Each of them would then be corrected in a priority like manner and on a one by one basis. This approach would prevent us falling into a vicious spiral which would cost us our lives. In addition, the specifications were written in such a way that it could be relatively easy to discard a proposal should it be the cheapest and in our mind not capable to do the expected work.

.../...

The intent of this manoeuvre was to prevent us from being forced to accept a pile of hardward which could not be proved that it would not do the job as defined in our specifications.

To this effect bench marks, accuracy of results, degree of responsiveness to our specifications, and others could have been used to eliminate the proposals of some companies whose salesmen were experts in presenting alternate ways to meet the intent of our specifications.

When this work was done two outside well known consultants were brought in to review our specifications. The idea was that no one could then question the plans and the design of our system from the outside or from the inside of our organization and should the system later turn out to be a success, no one could claim full credit for it.

In spite of everything, we experienced an unforecasted delay. A vendor which we knew had no chances to succeed claimed that he either received the request for quotation too late or not at all and consequently asked for an extension in order for him to submit a proposal which he declined at the last minute.

This delaying technique could have hurt us because, as every vendor knew, delivery was supposed to be right in time for the end of the fiscal year.

The acquisition of the system

The next milestone was the acquisition of the system whose hardware fortunately came in time and in one piece. The software, however, was not all available so we had to work with a previous release of the operating system for some time until the new one was shipped to us. This new release then was not quite acceptable and a few patches came later.

This delay was not a real problem because during this time we developed our own system modifications. However, when time came to accept the system we were greatly at the mercy of our supplier because the final responsibility to tie and make everything work was ours.

.../...

At that time, all the system software of our supplier was not yet available so we had to pay them on their good faith and it worked !

At this point there was no other way than to take this relatively small risk because involving a third party to take it for us would have cost us enormously and we would have lost control of our system.

Once this situation was cleared our protection was directly a function of the knowledge we had on our system and of the amount of energy that we decided to put into it to make it work.

Although all the above rules are basically perfect, they can be applied or followed only up to a point. The reason is that we cannot do everything ourselves and that what is effectively needed with respect to system software may require much more time than what was originally envisaged. This has nothing to do with the lack of planning or forecasting but simply that the accumulation of small run time problems may double or triple the expected time.

In addition to the workload, a specialist was brought in because we needed the intricate knowledge of the system in order to modify some system assembler programs with the assumption that one example should be enough.

Our consultant was working on a task basis mostly during odd hours and, as we discovered it later, made his programming in using hierogliph type macros, adding task when it was not really necessary so that his software turned out to be undebuggable and later, when we had more time to think it over we rewrote and changed approximately 90% of this program. This upgrade effort is never-ending. In order to be paid, the work of our consultant had to be acceptable. This is why we got a system which was fragile but workable. Although one task of the consultant was to document his work, he left suddenly without fulfilling the rest of his contract and obviously did not get paid for it.

Contradictors will say "you should never have done that!" However, only a small independent consultant could have accepted a contract on a task basis dealing with system software and including quite a lot of risks. On the other hand, well established companies were quite willing to lend us one of their programmers on a per day basis.

.../...

CONSUMER PROTECTION

The assigned programmer would have had some basic knowledge of our machine and he would have spent most of his time learning what we were willing to do. For our security and our benefit, this choice could not have been selected.

While our system software was being first made acceptable a large task was started and it was the conversion of a statistical package to our minicomputer based system.

This large project has been initiated by ourselves. We paid for the source code of the package without any documentation and with a fragile system, after more than six months of non-interrupted work, we produced the results of the first function. Obviously, the responsibility to convert the package was ours and with no outside help. The irony of this is that when most of the package was converted the original owner prevented us from reselling our conversion to other installations - even on a 50-50 basis - because we refused to let them have our changes for nothing. Although the package has been converted for our internal use, the above attitude confirmed that if we are not careful, we will have to pay dearly for what we need and give away what we produce !

What is recomforting in the data processing field is that the majority of people ignore that they are going to be robbed or exploited.

The conscious minority, which is requesting its due, has to be careful. Sooner or later, the word will spread that they are too tough in business and no one will be willing to do business with them. For those who have a limited budget the solution is unique :

1) be sure that you are not under-estimating the contemplated project

2) get as much as you can from your suppliers, and

3) be ready to fill in any gap yourself

.../...

CONSUMER PROTECTION

Finally, there is no such thing as a "taken for granted" protection. We can assure our protection in never accepting a loss and in trying to turn it into an asset. Our best protection is our knowledge and our ability to use them when we have problems.

REFERENCES

1. Datamation August 1977 Do-it-yourself Maintenance Management by the above author.

2. Datamation November 1976 Converting a Maxipackage to a Mini-system by the above author.