



*This Foreword was written for a reprint of Eli Goldratt's book "Necessary but Not Sufficient". I include it here because I continue to be dumbfounded by the observation that so many of IDEA's clients and prospective clients invest so much time, not to mention money, to implement technology that does not lead to dramatic improvements.*

*IDEA's product Elucidate™ quickly reduces investment and dramatically multiplies profitability. Yet, many times implementations are delayed for an ERP upgrade or a Transportation Management System installation. Simply, companies should work on the biggest and highest return projects first.*

*Thanks to Dr. Goldratt for providing this insight for our clients and prospective clients.*

*- Henry F. Camp*

## Technology – Necessary But Not Sufficient

By: Eliyahu Goldratt, July 2002

The In March of 1998 I was approached by an old friend of mine, Paul Baan. Paul and his older brother Jan founded, in the late seventies, a computer software company. Due to their relentless efforts and tons of business smarts they grew Baan into one of the world's leading companies in computer systems for organizations.

Paul was saying that even though business had never been better, for the first time it was unclear to them what they should do next. He asked me to analyze his company and suggest a strategy. My plate was full, but you don't say no to an old friend.

As I expected, it didn't take long to do the analysis. What I didn't expect were the results of the analysis, they were alarming. The analysis unequivocally showed that the entire computer systems industry was heading, like an express train, directly into a wall.

Companies in this industry were used to rates of growth of forty percent per year. If they had a concern it was how to find a way to leap frog their competitors - how to increase these rates of growth beyond the forty percent mark. My analysis showed that, very soon, some of these companies would go under. It also showed that it would be a fluke which one will be the first to tumble. Each company that went down would provide few more months to the others but not one would be able to maintain the traditional rate of growth. Moreover, within a few years everyone would be struggling to even make a profit.

I doubted if anyone in this industry would listen. The bonanza was too big, the profits were too hefty, and on top of it, the solution required a drastic change in the way these companies were doing business. Paul and Jan did listen and started the incredible task of getting the needed buy-in from the managers of their big and diverse company. But before the end of that year, before the buy-in process had been completed, the prediction of the analysis started to become reality. Unfortunately, BAAN was among the first to be hit. Badly. Today, four years after my first

conversation with Paul, forty percent growth per year seems a remote dream and there is hardly a single company in that industry that is not struggling to show a profit.

The validity of the analysis has been verified by reality. But, for reasons I will explain later, that is not enough for the software industry to adopt the solutions that are mandated by that same analysis. Not adapting the solution doesn't hurt just the computer software companies, it hurts their clients. In the last years almost every organization has invested a lot of money in computer systems (many invested tens of millions and some even hundreds of millions). In spite of those large investments I'm unaware of even a single organization that came forward and stated that its investment in computer systems had dramatically improved the bottom line. As a matter of fact, most organizations regard the investment in computer systems as a necessary evil. That is the biggest damage. Computer systems can revitalize organizations, can lift their performance to new levels. Provided that...

Provided that we will be able to answer the following questions:

### **1. What is the real power of the computer system technology?**

I believe that the power of computer system technology is in its ability to handle data. It has incredible power to store data, transfer data between silos and retrieve data. In each one of these three categories computer systems perform many orders of magnitude better than the technology we employed before, the paper technology. To prove that point, let's do a thought experiment (gedanken experiment). Imagine using the old technology to store your company data, which means please print all the data stored in your company's computers. Now, facing the resulting mountain of paper, search for one specific data element. How much time will it take? Compare it to retrieving a specific data element through a computer system. Most users, if they have to wait more than a few seconds, start to complain about a slow system.

No doubt, the power of computer systems is impressive. But let's not forget that not all managers in companies are technology freaks and most are rightfully interested in only one thing, in benefits, in the impact this technology has on their company's performance.

How can technology bring benefits? Only in one way: technology can bring benefits if and only if it diminishes a limitation. So what we actually have to do is to stop admiring the power of this technology and ask the next, disturbing question:

### **2. What limitation does this technology diminish?**

In my opinion, the limitation is: the necessity of any manager (in any level, in any function, in any organization) to make decisions without having all the relevant data.

Think about it. Remember that before computer systems, data generated in one silo was almost never available, in a timely manner, at another silo. From my experience I would not hesitate to say that for almost all decisions at least part of the relevant data is generated in another silo and, therefore, the decision has to be made without all the relevant data.

And, I'm not talking only about earth-shattering decisions. Take for example the case of a worker standing by a machine having in front of it some inventory. The foreman has to make a decision whether or not to instruct the worker to process this particular inventory. A vital data for such decision is whether or not there are significant clogs in the flow between this machine and the end customer. If there is such a clog, we know (from JIT and TOC) that it will be a mistake to now process that inventory. The worker should wait even if he has nothing else to do. If the clog is

outside the department of that foreman, what is the chance that he will be notified about it in time? The decision has to be made without all the relevant data.

In an ordinary organization, do you know about many limitations that are bigger than the one we are dealing with here? Bigger than: all managers are forced to make most decisions without all the relevant data?

A technology that diminishes such a huge limitation should bring enormous benefits.

But wait a minute. If that is the case, how come that we don't hear of many companies claiming that by installing computer systems they have ten-folded their bottom line results? How come we do hear about so many companies that are less than thrilled with their computer system?

Since it is apparent that computer systems usually do not bring significant bottom-line improvements, there must be something that is missing in our analysis. What is it?

Well, maybe we have to start earlier. We managed organizations before computer technology was available. How did we do it? It must be that long before the technology was available we developed modes of behavior, measurements, policies, rules that helped us accommodate the limitation (from now on I'll refer to all of them as just "rules" even though in many cases those rules are not written anywhere).

What benefits will we gain when we install the technology that removes the limitation, but we "forget" to change the rules?

The answer is obvious. As long as the rules that helped us to accommodate the limitation are obeyed the end result is the same as if the limitation still exists. In other words, we cannot expect to see any significant benefits.

So, it is vital that we be able to answer the third question:

### **3. What rules helped accommodate the limitation?**

In our case of computer system technology, the limitation is the need to make decisions without all the relevant data. The data that is missing is the data that is not generated in the local vicinity. No wonder that the rules that were developed to bypass the limitation are rules that helped to make decisions based on existing data, they are "local optima rules". Since the limitation existed for every manager, it is no wonder that we find these "local optima rules" in every corner of the organization (readers of my books are aware of plenty examples of such local optima rules in production, finance, marketing and project management and this book will point to many more).

Here is the place to highlight that identifying the old rules is not yet sufficient to determine the new rule. We, therefore, must proceed and ask the fourth question:

### **4. What are the rules that should be used now?**

In the case of computer system technology this was, probably, one of the most difficult questions to answer. For example, we all know that all of cost accounting is based on local optima, but what should we use instead? Some will say Activity Based Costing. I will say Throughput Accounting. But, how many of the computer systems are still providing the old "product cost" data? All of them, if I'm not mistaken.

How come?

Because many times the people who designed the computer system were not aware that some of the rules they observe are, in reality, an outcome of the limitation their technology is about to diminish. Due to that, they design the technology according to the old rules, and by that, cast the old rules in iron, damning the possibility of their technology bringing real benefits. This, in my opinion, is exactly what we witness regarding computer system technology. This is the reason why software providers talk about "better visibility" rather than about startling bottom line benefits.

To make computer systems bring what they are definitely capable of delivering, a huge jump in organizational performance, we must proceed and answer the next question:

## **5. In light of the change in rules what changes are required in the technology?**

In the case of commercially available computer systems my estimate is that we have to replace about 1-2% of the code. And we should erase about an additional 30%. I hope that within the next years we'll see more commercially available systems that are based on the new rules. As for the time it will take until the redundant code will be erased I'm much less optimistic.

And then of course we still have to answer the biggest question of them all:

## **6. How to cause the change?**

We all know that changing from an old technology to a new one is not simple. Now we realize that changing the technology is the smallest part of the challenge. To get benefits we must, at the same time, change the rules - rules that are cast into modes of behavior, into culture.

This is probably the reason for the reluctance of most software companies to push systems which are based on the new rules. They rightfully consider their companies as not qualified to change the way organizations are managed. Talking to many of the executives of software companies it is obvious that they will rush to satisfy whatever the market demands. So, the key is in the hands of the individual organizations.

What is needed is that enough companies will realize that if they want to succeed they must address their biggest constraint. And right now, the biggest constraint that most companies face is the fact that so many of their rules are based on devastating local optima.

Eli Schragenheim and Carol Ptak, my gifted co-writers, convinced me that the best way to guarantee that this message will have an impact is to write the book as a novel. In this way the readers can familiarize themselves with the inside story of all parties involved, the software companies, the integrators who implement the software and most importantly the dynamics in an organization around an implementation of a computer system.

A technical book written in the novel format carries with it some risks. In a technical novel, mistakes, or even things that are just not explained well enough, stick out like a sore thumb. Any reader, even a complete novice in the subject matter, spots such weak points and relates to them unrealistic or not true-to-life. Three or four such weak points are enough for most readers to put the book down with disgust. Therefore, writing a technical novel requires bringing all the information to the level of perfect clarity. But then, so many of the readers, even though they enjoy the book, relate to the content as "just common sense." That by itself is not a problem, the problem is that since it is "just common sense" many readers ignore the information and continue to follow the existing common nonsense.

I hope that you will read this book, enjoy the plot, think about the content and if you find it to be "common sense" I do hope that you will not ignore it but rather implement it.