

Viable Vision

By Dr. Eliyahu M. Goldratt

Maybe the best way to describe what I mean by a **“Viable Vision”** is by a quote taken from a letter I wrote to my friends in November 2002: “When I do an analysis of a company I am somewhat satisfied only when I clearly see how it is possible **to bring the company to have, in less than four years, net profit equal to its current total sales.**”

Knowing the response of most people to such a claim, my next sentence was: “I also learned not to share this expectation with the top management; they will take it as a decisive indication that my suggested solution is unrealistic.”

During 2003 I put the above to the test; I put to the test the reaction of top managers. Rather keeping the vision to myself, I put it at center stage. Not just the vision but, mainly, exposing the reasons for my conviction that that incredible vision is viable. I started by sharing the diagnostic of what is currently blocking the performance of the company. Based on that I deduced, using solid cause and effect logic, the tangible steps that are bound to remove that block. Then I dived into detailing the steps that have to be taken in order to capitalize on that breakthrough; the steps that will carry the company into having, in less than four years, yearly net profit equal to its current yearly sales. Done in this way the first reaction of the top managers was: “This is just common sense, **why haven’t we done it before?**”

Why haven’t they done it before? How come that the prevailing notion is that, unless the company has a unique product or unless the company is very small, it is unrealistic to expect that a company will increase its net profit by so much? How come that even though it is possible to construct a Viable Vision for more than half the companies, the prevailing notion is that it is impossible?

The answer is that most people are unaware of the fact that **any complex system is based on inherent simplicity. And capitalizing on the inherent simplicity is what enables incredible improvements within a short time.**

What is “inherent simplicity?”

To explain this concept we first have to clarify what do we refer to as a “complex system.” The more data one has to provide, in order to fully describe the system, the more complex the system is. If one can fully describe a system by four sentences, it is a simple system. But if one needs thousand pages to do it, the system is complex. How complex is the system you are working for? How many pages are needed to describe every process on every part, the relationships with each client, etc? It is no revelation that companies, even the small ones, are extremely complex. It is also no revelation that it is difficult to manage a complex system.

So how do we go about managing a complex system? We dissect it to subsystems, each is, by definition, less complex than the whole. If you have any hesitation to accept that that is precisely what we do, just look on your organization chart.

Dissecting a system into subsystems has its price. It leads to mis-synchronization. It leads to harmful local optima and, in some cases, even to the devastating silo mentality. Since our systems are incredibly complex, it seems that what is left to do is just to minimize the price; to do the best we can to improve synchronization, to foster better collaboration between the subsystems.

As long as that is the only option that we consider we'll be under the impression that achieving a significant jump in profit within a relatively short time is a rarity. We'll be under the impression that "bringing the company to have, in less than four years, net profit equal to its current total sales" is unrealistic.

To see the true potential of a company one has to delve deeper into the issue of complexity. What bothers most of us is the fact that part of the data that typifies our system is not relating to just one component of the system but to the relationships between two or more components. In other words, the thing that makes our system more difficult to manage is that what is done in one place has ramifications in other places. That cause and effect relationships turn our system into almost a maze. But that fact is what provides the key for the solution.

Think about it in the following way. Examine a given system and ask yourself, **what is the minimum number of points one has to impact in order to impact the whole system?** If the answer is "ten points" this is a difficult system to manage, it has too many degrees of freedom, it is like attempting to manage a bunch of wild cats. But, if the answer is "one point" then this system has only one degree of freedom, it is an easy system to manage.

Now, do you agree that the more interdependencies exist between the various components of the system the less degrees of freedom the system has? Considering the enormous complexity of your system it stems that there are very few elements that can be used to govern the entire system. In other words, the more complex the system is, the more profound is its inherent simplicity.

To capitalize on the inherent simplicity we must be able to identify these few elements that govern the system. If, in addition, we are also fully aware of the cause and effect relationships between these elements and all other elements of the system, then we'll be able to manage the system to achieve much higher levels of performance.

These few elements, being the ones that dictate the level of performance of the system, are the constraints of the system. That also implies that the constraints are also the leverage points of the system. Hence the name I choose for this approach – the Theory Of Constraints - TOC.

The process to capitalize on the inherent simplicity is straight forward:

1. Identify the system's constraint(s).
2. Decide how to exploit the system's constraint(s).
3. Subordinate everything else to the above decision.
4. Elevate the system's constraint(s).
5. If in the previous steps a constraint has been broken go back to step 1.

It doesn't matter what system you address, when you approach it through its inherent simplicity the results are always the same, a surprising jump in performance and the impression of "This is just common sense, why haven't we done it before?" Twenty years ago, I demonstrated it on production systems (plants) in my book *The Goal*, then on projects-based systems in *Critical Chain*, on marketing/strategy of companies in *It's Not Luck*, and three years ago, on a whole industry in *Necessary But Not Sufficient*. In each case, the many companies that followed it had validated the results.

Still, most managers are oblivious to the concept of inherent simplicity. As a result, they are still looking for sophisticated and complex solution. They still don't comprehend the magnitude of improvements that are within their reach.

In one of my discussions with Carol Patk, VP of PeopleSoft and my co writer on *Necessary But Not Sufficient*, we ponder on this issue. What can be done to awaken managers to the treasure that lays here? Carol suggested that the best way is to give people a chance to experience the concept first hand. Since even small systems are still complex, let's present managers with a simulation of a small enough system that they can test and re-test the results of managing it in different ways. "Let them realize," she said, "that using all the sophisticated tools that they have will result far from satisfactory results. Moreover, since we'll use a computer simulator they will feel free to deviate from their cast in iron procedures and local measurements, they will be free to use their healthy intuition." Using healthy intuition is bound to lead people to realize, first hand, the validity of managing according to the system constraints, to the power of capitalizing on the inherent simplicity of a system.

Hopefully, once people will experience it on one example they will be more open to entertain the notion of even a Viable Vision.

And that's how this competition was born. I do hope that many of you will grab this opportunity to expose yourself to a common-sense way of managing systems.