

## CHAPTER VI

# REENGINEERING THE USER ORGANIZATION

### Strategic Objectives and Reengineering

There is nothing permanent in strategic goals. In the 1960s the height of fashion in business was corporate diversification through mergers and acquisitions (M&A). Size was regarded as a criterion of excellence, and the drive was to grow fast.

In the 1990s, this was followed by the downsizing strategy and *reengineering*—the turning of the organization inside out with the dual goal to:

- slim down and
- find new synergies *within* the enterprise.

Focused business came into fashion in the last decade of the twentieth century, with some experts advising that diversification had done more to destroy shareholder value than to create it. Even former fans of giant corporations started to praise smaller firms. Having helped to train generations of middle managers in how to leverage, some companies practiced de-leveraging as a strategic decision.

All that did not last long. Leverage took off with the dotcoms in the late 1990s. After the year 2000 bust, both trends—“smaller is beautiful” and “leverage hurts”—have been reversed. The result has been the 2007-2009 deep economic crisis, where big banks fell on their swords, and it took an inordinate amount of public debt to pull them out of their self-engineered descent to the abyss.

With this, the concept of reengineering is back in fashion, appealing to executives who adopted the policy of rethinking and redesigning their enterprises and IT systems.

Well-governed entities developed new information technology paradigms as well as basic business practices like product development and *customer service, and other lines of business required quick execution. Reengineering brought improvements in productivity, quality, cost reduction, and customer satisfaction.* Some companies, however, missed the opportunity to regenerate their systems and procedures. Either they stuck with their crumbling structure or the reengineering job they did was ill-directed or half-baked.

Companies that missed the opportunity to profit from new developments held the wrong belief that throwing money at technology would produce miracles. That's a false premise. Technology is at the service of reengineering, which comes first, not vice versa.

Successful reengineering projects require clear-cut top management decisions as well as execution skills. The lack of either or both has been a stumbling block to every initiative to gain the upper ground in competition. The challenge of corporate renewal is so much greater when we talk about reengineering in connection to strategic inflection point—an example being the advent of cloud computing.

With cloud computing reengineering is not an option. It is a prerequisite. Migrating data from legacy applications into the cloud is a tough job, and it is not a one-time task either, some pundits say.

Though specific issues vary from one company to the next, the way to bet is that the amount of effort for streamlining systems and procedures will be significant and the ability to do so should be a basic factor in deciding to use the cloud. True enough, this task is already spawning new opportunities for firms providing deep integration across public and private cloud but relegating the mission of reengineering to insourcers without the company's own personnel playing an active part in it is highly inadvisable.

Not all projects in restructuring and reengineering have been successful. In many cases money has been thrown at the problem—evidently with no results. That many reengineering efforts fail seems depressing—the reason being that not all CEOs and CIOs appreciate that success or failure is determined by:

- a company's understanding of the process and its demands and
- the ability to execute reengineering in a way that the whole firm is committed to it.

Achieving synergy requires an ingenious sales, rather than flat orders, effort coupled with the determination to overcome hurdles and show results. The primary causes of reengineering failures fall into four groups. One is what General Maxw Taylor called “the uncertain trumpet”: *if* the board and CEO do not know *what* they want to achieve through the cloud, *then* nothing will be achieved.

Whether or not cloud computing services should be used is a strategic decision and the first to be made. Benefiting from onDemand software means abandoning the (wrong) past policies of massaging packages. Therefore, makes sense to spend time and energy on analysis of the current organization. Sin reengineering entails discarding current process designs, plenty of procedural der, is destined for the rubbish heap—the challenge is to decide which. It is easy to £ victim to a protracted analysis, postponing that fateful day when sweeping chans have to be made.

### **Organizational Solutions Are No Sacred Cows**

Any reengineering project targeting cloud computing, whether this concerns the internal or external cloud, must pay great attention to the fact that all sorts of applications are data intensive and continue becoming even more so. Not only does this significantly complicate reliability and security associated with data placement, access, and transport, but it also calls for the greatest care associated with:

- size of data storage and
- data traffic at nearly every level of the system functions.

If after adopting computing the company intends to continue expanding stor-age requirements at the currently prevailing levels discussed in Chapters 4 and 5, *then* surely what it needs is *not* a rush to the cloud. If I were given a choice, my first, second, and third choices would be reengineering and restructuring data storage needs at every layer of the organization without worrying about the cloud but con-centrating on a thorough and deep restructuring.

These ever-growing trillions of information elements and files don't exist in the abstract or by an act of God. They are connected to departments and functions in the organization. That is exactly where the reengineering study should start making sense, by combining them and simplifying them. This should be done fast, like in an audit.

- An IT audit challenges not just cost but also, if not primarily, lines of authority and responsibility—hence organizational solutions.
- Organizations are made of people; as people and functions change, chosen solutions become obsolete or outright wrong.
- The principle is that no organizational status quo or its details are a sacred cow to an audit and therefore to reengineering.

When a company asks me to conduct an audit of its technology status, my bet is that there is more than a 90 percent chance it has allowed both its headquarters and its operations worldwide to maintain separate, functionally duplicative systems supporting operational and reporting requirements for its various business components. This has two effects:

- it multiplies its files by a factor  $n$ , sometimes an order of a magnitude, and
- it makes up a highly heterogeneous environment in collecting, analyzing, and processing data.

Another educated guess dictated by experience, and supported by the facts, is that the company's management has asked for an audit if that consolidation of information elements never took place. Maybe the CEO or CFO asked for it, but never got it. Subordinate business units feeding information upstream—a job quite often done manually in spite of huge amounts of money spent on IT—just did not care to execute orders from headquarters, and they found the political clout to do away with orders from above.

Duplication and triplication of files cost dearly in expenses and delays. Even if costs are left aside, there is the problem of accuracy in stored data. Under these conditions, their consistency and accuracy become practically impossible to control—a situation that should definitely be corrected through reengineering prior to joining the cloud.

Many people say that the need to reengineer a major part of the company's database infrastructure is not self-evident. This is wrong, but it explains why one of the key questions facing companies today, to which there are no easy answers, is How long after a thorough restructuring of information elements and files should we go ahead with a new reengineering effort?

There is no universal answer to this query. The IT audit should provide a factual reply, within the realities of the environment within which it takes place. The only general remark is that practically every company today has, at least in some part of its operations, legacy data processing from the 1970s and even the 1960s.

- These have exhibited a crying need for restructuring, but typically the decision has been repeatedly postponed.
- With outsourcing infrastructural chores to the cloud, however, IT's inflection point works as a catalyst to finding what it takes in data organization.

There is *simply no* alternative to making the *organization of the company's data-base state of the art* through reengineering: weeding out duplicates, setting and using file standards, and making it integrated, highly functional, reliable, flexible, accessible in a seamless way, and able to be operated at reasonable cost.

Neither are all data on storage readily accessible and easy to find. With legacy software, data retrieval procedures are time consuming and not user-friendly. There may also be massive ongoing errors that should most definitely be identified and corrected through reengineering. *If error* sources are left as they presently stand in a cloud environment, *then* they will paralyze the user organization and bring its operation to a halt.

In addition, to provide a dependable picture in respect to data sharing requirements in a multisourcing deal, the reengineering effort must determine how the vendors' different application programming interfaces (APIs) are used in connection to cross-platform information elements. Inevitably, this impacts the company's IT environment.

Internal company politics, too, must be handheld in an ingenious way. It is not unlikely that those doing the reengineering work will encounter negative reactions, even by people who profess that cloud computing is the way to go. Entrenched interests may not only be reluctant to accept a downsizing of their systems and procedures (and budgets) but also try to prolong to infinity the reengineering work.

- Company politics are present no matter what one is doing; that's not news to people who have tried to improve organization and structure.
- Precisely for this reason, I have underlined that *//* reengineering does not get 100 percent support by the board and CEO, *then* cloud computing will end in cloud cuckoo land.

Compromises lead nowhere. A project that is not well done is no project at all. In a well-done reengineering project descriptions of processes and data models address the conceptual and logical dimensions of enterprise information, trans-forming data formats prevailing in diverse and incomparable legacy systems into a searpless virtual environment.

The complexity of the job may be increased by the unique characteristics of the user organization's legacy system(s). It is, however, proper to remember that with the Y2K challenge of the late 1990s, those companies benefited from the invest-ments they made to use it as an opportunity to reengineer their IT—and sometimes the firm.

Another issue to be kept in mind is that there is no such thing as an *a priori* suc-cess, as some people are professing. But there are *a priori* failures, when the project starts on the wrong foot. In the nearby six decades of computer applications, there has been a swarm of IT project fiascos:

- nearly a third of new IT projects have been *cancelled* before completion, at an estimated combined cost of billions, and
- more than one out of two projects that have been completed were almost 200 percent or more over budget, at an additional cost of billions.

These cases occur regularly in all sorts of companies and all in-house IT proj-ects, and they are due to many factors: lack of senior management involvement, making projects vulnerable; fuzzy project goals, sometimes because of compro-mises; lack of reengineering; weak composition of project teams in a skills sense; lack of leadership; and senior management's acceptance of steadily slipping time to completion and escalating project costs. Keep all that in mind when thinking about cloud computing.

## **The Number One Asset Is Human Resources at the CIO Level**

Nothing can be achieved without first-class human capital; therefore, the people put in a project should be first-raters. But this is by no means true only of IT projects) A study done by Bankers Trust in the late 1980s, at the peak of its night, documented that the greatest assets of a bank are not the money and other wealth in its vaults but people and people. People its clients and people its employees. I should have rather written “well-trained people,” which is a never-ending process because knowledge wears out. It decays if we don’t use it. With this in mind, some years ago John F. Kuemmerle, senior vice president/administrator of Guardian Life Insurance Co., remarked that computer people’s skills can quickly become obsolete and, to counter that, they should spend a good part of their time learning.

Few companies follow this policy, yet without it the implementation of advanced solutions is going to be defective at best.

Take onDemand software vs. onPremises software as an example of what needs to be relearned in a different way than in the past.

Traditionally, the policy of company-developed and -owned software allowed the information technology people to deploy applications specific to the ongoing business through custom-made and customized code. The former was developed wholly in-house, and the latter by altering purchased applications packages. A timetable of two to three years was a classic in either case. Quite often,

- bought software was changed so much that at the end it had nothing to do anymore with the vendor’s version, and
- this meant a great expense and significant delay in implementation, along with the impossibility of assigning the maintenance to the vendor.

What has been part and parcel of traditional IT culture is no more possible with onDemand software bought from the cloud. OnDemand applications are on rental, and the code sits external to the user organization. Such application routines are shared with a number of cloud customers who will not take it kindly if one of the other users tries to change something (even if it could do so)

## Promoting Greater Productivity through Reorganization

Computers and communications, as a system, must be learned not in the sense of machines but rather in that of a generic aspect of industrial and cultural life. And it should be updated fast enough, as the facts of real life rapidly change. True enough, the traditional approach is to look at IT as an introverted subject reserved to the system specialist. This is, however, deadly obsolete, and therefore the wrong way.

Technology is an *enabler*, it is not a goal, just like an airplane is not an objective. Computer technology is a *means* for doing specific processing jobs; likewise, the airplane is a means for transportation. Before boarding an airplane we must decide where we want to go. The same is true when we are using technology—new or old. What do we wish to reach?

Say that our goal is to improve the productivity of our firm. Can cloud computing help in this direction? To answer this query in a factual manner, we should return to the fundamentals, looking at what promotes productivity. Such promotion has four components:

- speed of innovation,
- pace of capital investment,
- quality of the workforce, and
- a variable driven by the business cycle.

The first three have longer-term impacts and are the real determinants of ongoing productivity growth under direct control of senior company management. By contrast, the fourth component is exogenous and its impact is shorter term, following a sinusoidal curve.

- In an expansion, productivity takes off as firms use their existing staff more effectively before hiring new workers.
- But as an economy slows, companies don't fire workers immediately. Hence productivity slows.
- Then starts a sharp increase in unemployment and (in a way, surprisingly so) productivity starts taking off.



This mismanagement of information has much to do with lower productivity as people spend time in trivia and in searches for documents. Indeed, in spite of huge IT expenses, companies continue producing useless amounts of paper.

Inefficiency in document handling is a general rule, but it is unevenly spread in an entity. Some parts of the organization are worse violators than others. In a bank, the back office provides an example of urgently needed IT restructuring. One of the major reasons for falling behind in the use of technology is the split in emphasis:

- front desks get real-time support, while
- back offices remain with medieval technology.

Theoretically, but only theoretically, efficiency in the use of physical resources, such as IT gear, is not the same thing with the human resources productivity. This is untrue, however, because the two subjects correlate. The same statement is valid for efficiency and effectiveness.

It is therefore a legitimate requirement on the side of user organizations that the cloud computing vendor demonstrates how efficiently it uses the physical resources that it provides as a service; for instance, how efficiently the cloud vendors disk storage is being used.

Today, in the average among IT shops, only 6 percent of server capacity is efficiently used; another 30 percent contains garbage nobody bothered to remove. Cloud computing providers are most elusive in answering questions on how they will be improving these ratios without reengineering the client company and its legacy information technology solutions. Promises have short legs. The user organization:

- should ask for facts and
- should have them verified through a hands-on examination.

Restructuring, and not the label *cloud computing* per se, addresses the reason in the last bullet. It is normal that the user organization should expect significant improvements by way of a more sophisticated use of IT, but these will not come through labels or wishful thinking.

## The Transition from Legacy to Competitive Systems

If cloud computing is chosen by a user organization as its IT solution, whether this decision centers on onDemand software and platforms or extends into infrastructural services, plenty of challenges will face the CIO in the transition from legacy system(s) to the new environment. This is indeed a demanding task, most particularly so if the user organization wants to migrate its database (in whole or in part) to the cloud infrastructure. Because of this, the project must be

- properly planned,
- executed with an iron hand, and
- effectively controlled in all its aspects: quality, cost, and time being at the top of the list.

This transition should be seen as an integral part of reengineering, because it meshes with reorganization and restructuring in practically many of its aspects. Details on how to proceed vary from company to company, and for this reason it has been a deliberate choice to treat the process of transition through a generic approach leaving to the user organization the task of adapting it to its own circumstances.

Cloud vendors who bet on old technology, such as the fifty-year-old main-frames, are miscalculating because cost savings will be an illusion and bad news has the nasty habit of spreading fast. When it does, the evolution of cloud computing from proprietary platforms will slow down to a drop, as evidence will show that the heralded added value is a fake.

It is bad enough that the mainframe environment is still around after decades of architectural changes and breakthroughs in service-oriented solutions unsupportable by old concepts. Other than conflict of interest, there is really no reason why we should make matters worse by bringing mainframe mentality into the cloud.

## Avoiding the Medieval EDP Mentality

CEOs who are in charge of the company under their watch present a challenge to the old regime of IT by demanding that new solutions are responsive to new realities. A user organization will be lying to itself if it believes that the transition from the legacy to the target system is something painless—whether a public or private cloud computing environment. There is one exception to this statement: *if* the conversion is from mainframe mentality to mainframe mentality, *then* the job would look easier (albeit not always so); however, the benefit will be zero and return on investment nil.

It is not only conceivable but also very likely that some cloud computing vendors will try to get the contract by downplaying the need for cultural, structural, and system changes. They will do so to calm the CIOs and his or her cohorts' fears that they have plenty of hard work to do (and if they fail they may lose their job). There is plenty of evidence of this strategy, based on patches—just as there is plenty of evidence that patches fall apart.

The best strategy for CIOs who want to avoid patches and the risk of failure is to look for and find a wholesome onDemand solution, replacing the old incompatible and unmaintainable programming structures with a new and flexible system. The good news is that a dozen years ago Y2K provided excellent precedence to this well-rounded approach to the renewal of a company's programming library. Hence, prior to deciding anything, it is advisable to:

- visit the organizations that have year 2000 experience and
- learn from them, particularly what went wrong and how they corrected it.

In parallel to this, the reengineering/transition program must ensure that all required changes to data formats and data files are done in a serious, comprehensive manner. In this connection, the way to bet is that it will also involve software at the microcode level. To which extent the latter needs to be done will depend on both the present and target systems. No matter what the challenge, data conversion must be reliable and timely.

A wise CIO will never be thrifty with precautions. If the cloud computing vendor suggests electronic file transfer, then the user organization's answer should be that this is nearly the last step—not the first. Methods for the early steps, but for an interim, not permanent solution, are data mapping and data conversion.

- Data mapping works through declarative descriptions written in a domain- specific language.
- These describe how data are mapped from the legacy to the target system

This is a rather aged method for data conversion but is still being practiced, and it requires two stages. Compiled by the converter, data mappings cause the new applications database to be updated. A browser will show the relationship between *J*information elements, representing the mapping between legacy data and target data. (The alternative had been an emulation approach, which made a PC act like the screen of a stupid terminal with 3270 protocol. This evidently added to the costs and made a mockery out of the power of microprocessors while graphical user interfaces remained rudimentary.

Another folkloric version of “solutions” pushed by mainframe vendors, and dis-tinguishing itself by falling back to medieval EDP mentality, has been emulation using screen scraping—replacing a terminal screen with graphical representation of data. All counted, this and other approaches still in use add so much disk storage to the mainframes that they lead to system bottlenecks. Critics would say that the logic of all that is wanting, and experience documents that they are right. However, there is a lack of efficient solutions in that sort of conversion.

The only efficient alternative is to redesign and recast the database—a major job. Moreover, many IT people fail to appreciate the contrast in efficiency between old technology and new technology; yet the difference is striking.

- the client-server architecture is too distributed and a waste of network resources, and
- the best policy is to view the web and Java as a way to return to the central-ized approach.

No technology is in itself good or bad. It all depends on what people and companies make of it, and what experience teaches is not positive for mainframe solutions.