Managing Performance

Geary Rummler & Alan Ramias
Consultants
Performance Design Labs (PDL)

GRummler@ThePDLab.com
ARamias@ThePDLab.com

IT Disasters: A Root Cause

In our first column we wrote about the huge, costly failures of IT systems development projects, citing some of the disasters we saw personally or had read about recently. But we figured there were more – bigger and badder than any we had uncovered in our decidedly unscientific research. And, sure enough, an alert reader wrote us to point out two more appalling cases of our tax dollars being burned away:

- One was an attempt to replace the aging FAA air traffic control system, stopped after $4.5 billion had been spent, of which only $75 million was salvageable for local airports. The project was then restarted, and the current estimate of costs has grown to $51 billion (from an initial estimate of $12 billion).

- The other was a project to modernize the IRS system to process tax returns. Started in 1985 and running for 10 years, the project was stopped. More funding was then obtained, and another effort began in 1998 to process simple tax returns – for a cost of $8 billion.

These incidents, and others, were also cited in a recent book, *Dreaming in Code*[1], that attempts to elucidate – from a programmer's viewpoint – why these things continue to happen. The viewpoint that interests us, however, is that of the business leader who, after all, is paying for these failures.

We believe that a big part of the failure to choose and implement technology effectively is traceable to not understanding the nature of the organization in which the technology is to be used. The purpose of technology is to enable human performers to do their work or to replace manual work with automation. What is being enabled or enhanced are business processes. So it follows that if an organization's business processes are not well defined, designed, and managed, then the technology that enables them is going to be an imperfect fit at best. How can rational choices about technology be made without a deep understanding of how work is performed in a given organization and how best to monitor, manage, and guide that work?

Managing the Organization as a System

The fundamental tenet of our work with clients is that every organization is a complex system of inputs, business processes, and performers, converting customer needs into valued goods and services, as depicted in Figure 1. The best way to understand what goes on inside any organization is to identify its processes, each of which consists of a sequence of work activities that progressively add value until a valued output is produced.

This view is in contrast to the classic functional view of organizations, in Figure 2, in which work is viewed as occurring in organizational silos of functions and jobs, and there is little or no recognition of the interdependencies between jobs and functions needed to accomplish the work. The outcome of this view of organizations is that work gets managed primarily within the

[1] Rosenberg, Scott, *Dreaming in Code*, Crown Publishers, New York, 2007.

functions, leaving unmanaged "white space" between functions, and between the types of work (or processes). The emphasis is on management of resources expended within functions, and the potential for competition over resources is at a maximum, while the critical objective of delivering value to customers – which should drive the entire organization's goals and actions – is obscured.

Understanding an organization as a system is the starting point for correct decision-making about technology. Our framework for helping a management team – or enterprise architects, or software designers – applies this concept of an organization as a system to a given decision (such as, "Should we buy this new technology?", "Should we upgrade this or that application?", "Should we ask employees to use this tool instead of that one?"), which is the Processing System Hierarchy shown in Figure 3. This is a top-to-bottom framework for organizing work, or analyzing work being performed, or deciding whether to enhance work being performed. Inside this framework are levels of processes that can be decomposed into smaller and smaller components until you reach the individual performers and their enabling tools and technologies. Note that this decomposition addresses smaller and smaller units of work, but does not decompose into functions, as functional decomposition inevitably leads to silo thinking and silo management. (Of course, the ultimate purpose of this multi-level hierarchy is to create and deliver value to customers, so we could also call it a "Value Creation Hierarchy.")

Level 1 – Super System

The Super System Level provides a view of the entire enterprise as a system. This is the appropriate starting point for decisions about technology, although it is rare that technology projects start at such a macro level. In a vertically-managed organization, even an ERP system implementation will take place within silos instead of across the enterprise.

The Super System view depicts (1) the external variables that exist outside the control of the business but which affect how it operates, and (2) the internal business model by which the organization provides value to the market and how it makes money in providing that value. If defined effectively, the Super System view should articulate strategic direction, strategic priorities, and performance requirements for all levels of the enterprise.

As the highest level of the Processing System Hierarchy, the Super System view is a great starting point for working with executive teams and others to develop an Enterprise Architecture, to formulate or validate an organization's strategy, to identify, diagnose, and improve business process architectures and key processes, to sketch out future-state scenarios, and to identify the implications on the company's processes, technologies, people, and infrastructure, and to help clarify an organization's business requirements that should drive technology requirements.

Level 2 – Value Creation System

Here is depicted what we call the Value Creation System (VCS), where we make visible at a high level the value (i.e., the goods and services that customers want) that is created, promised, manufactured, and delivered. To provide value, a business must do three things:

- Identify the customer's needs for a product or service, and then find, invent, or design something to meet that need. This involves processes we commonly think of as research, product development, product launch, and product lifecycle management. The major output is the right product/service as reflected by sustained sales and profitability.

- Make customers aware of their need for the product/service and convince them to buy from the business. The output is the identification, capture, and retention of customers.

- Deliver the product/service to the customer. This typically includes all those processes involved in order processing, manufacturing, delivery, installation, invoicing, receiving, and warranty management. This output is the efficient delivery, installation, and maintenance of the product/service.

Level 3 – Primary Processing Systems

Inside each component (Launched, Sold, Delivered) of the VCS is what we call a Primary Processing System (PPS). Each PPS in turn is made up of the work processes necessary to produce the outputs and achieve the goals of that given PPS. Although, as you will see, we are fairly adamant about the three Primary Processing Systems that constitute the VCS, we don't hold that there is a generic list of processes for each PPS. For example, the Launched PPS will necessarily vary by industry (retail, petroleum, electronic, consumer goods). At the level shown in Figure 3, the three processes making up the Sold PPS probably hold for most businesses (until you go down to the next level).

At the Primary Processing System level, *it is very much about the unique organization of the value-adding work of a business so that it can be effectively performed and managed and provides the potential for a competitive advantage*. As a result, process definition (i.e., the organization of work) should be driven by the specific strategies, goals, and competitive challenges of the individual business as you move down the Processing System Hierarchy.

Level 4 – Process level

In Figure 3, the process level example is an explosion of the "Order Obtained" process shown in the "Sold" Primary Processing System. Again, the steps in what is essentially a sales process should be unique to a business, given its industry, customers, and products. But note the following:

- At this level there is a *growing potential for conflict between unique requirements of a process versus available off-the-shelf software applications.*
- In the case of the example in Figure 3, the unique requirements of an effective sales process could be greatly compromised by the installation of an off-the-shelf sales process and management system software application.
- Software packages offered at the process level can lead to unfortunate hard wiring and the premature "hardening of the arteries" of an organization's Processing System Hierarchy.

Level 5 – Subprocess/Task/Sub-task

The Subprocess level is the link to the performer, through the structure of work tasks and sub-tasks. An individual, a technology system, or a combination of the two ultimately performs the work of the organization. It is here where technology developers tend to dwell, and, while they may be doing their utmost to provide effective support to performers, they can be operating with blinders if they lack the larger context of Levels 1 and 2.

If requirements, for example, are built starting at Level 5, they inevitably reflect the views of the requesting organization, which is usually a functional area, not the enterprise. The view here is also a narrow view of work, because the true scope of work that produces value is at the VCS level. To focus at this level or below is to risk sub-optimization because the customer is no longer particularly visible, nor are the whole flow and interdependencies of work.

So you can see we are recommending that this Processing System Hierarchy framework be utilized top-down by IT to understand, validate, and prioritize business needs, trace those needs

down through the hierarchy to the appropriate level and scope of processes where change must happen, and then – and only then – to design technology solutions to fulfill the business needs.
The risk of not understanding the organizational context for a technology decision and just jumping to the level at which a need is first expressed (for instance, at the level of an individual process or a performer) creates the potential to make changes that will be misaligned with the needs of the business and thus may unintentionally sub-optimize organization performance. That is assuming the technology even gets built. In the worst-case scenario, the technology prescription is so ill-advised that the project blows up somewhere in the middle of the muddle.

Implications for IT

For programmers and developers working on particular enhancements, it may seem to be a daunting task to define and understand an organization's entire VCS if all one wants to do is decide whether to design an application or not. But add that individual decision to many others – all decided in isolation – about potential uses of technology to improve work, and it can add up to vast amounts of wasted time and money. And, in truth, it's not the people inside IT who should be defining the VCS; it's their leaders, in concert with business leaders, who should do so. A defined VCS should be *available* to IT specialists, not created by them.
IT leaders are ultimately the ones holding the bag for major IT disasters, even when the impetus for some large-scale initiative that derails was not their idea to begin with: They are expected to provide the foresight and guidance that helps the executive staff make the right decisions about technology. Using the VCS as a framework for decision-making about technology can go a long way toward helping them be effective strategic advisors.

In Future Columns

The lack of understanding of the organization as a system is not the only root cause of IT disasters, but we are convinced it is a major cause. In future columns we will elaborate on a couple of others.
Then we will add more to the framework we described here and its implications for better IT-business alignment. And in that context we will also describe the anatomy of an IT disaster and offer some ways to avoid such a scenario.

Figure 1

The Organization as a System Lens

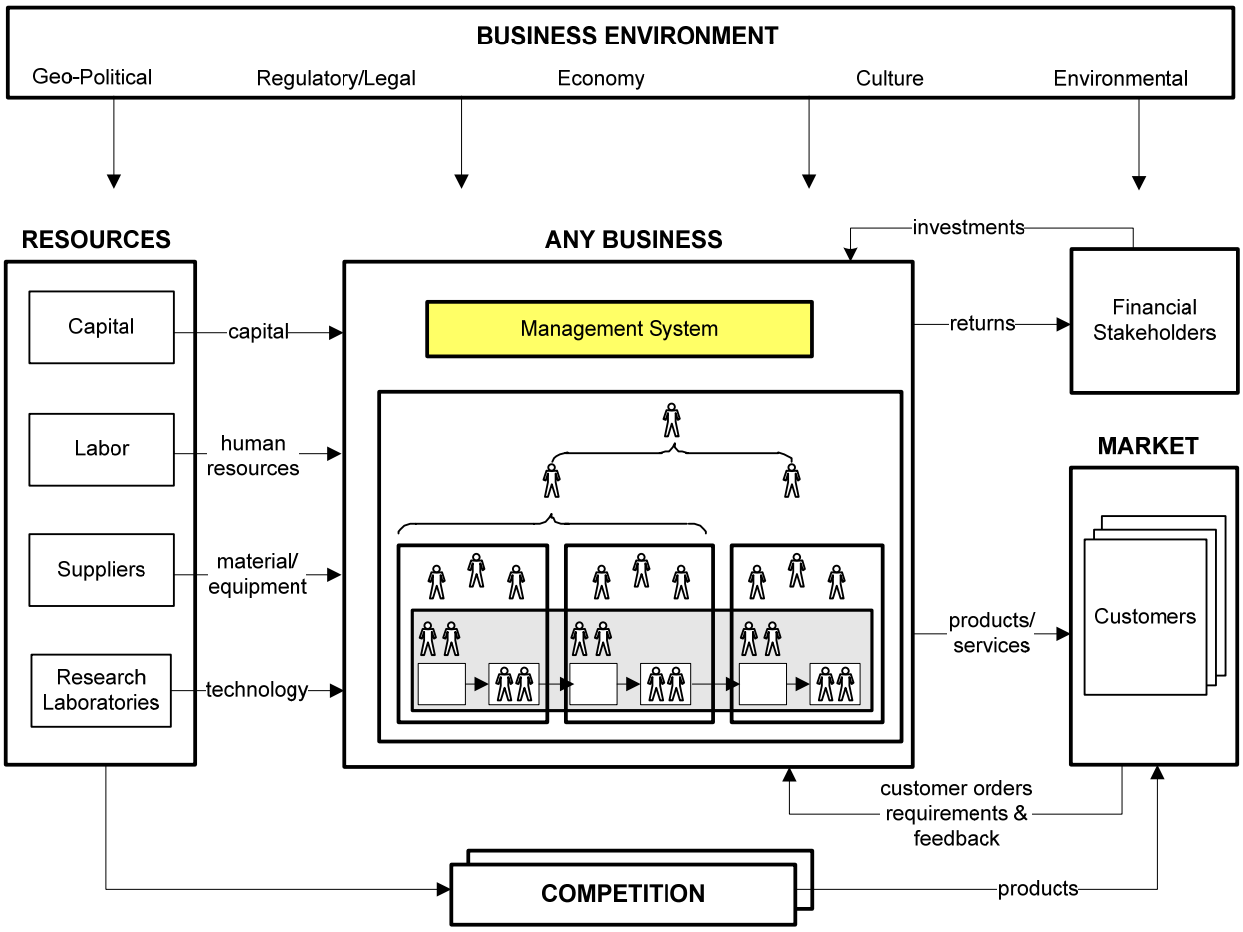

Figure 2

Organization Chart View

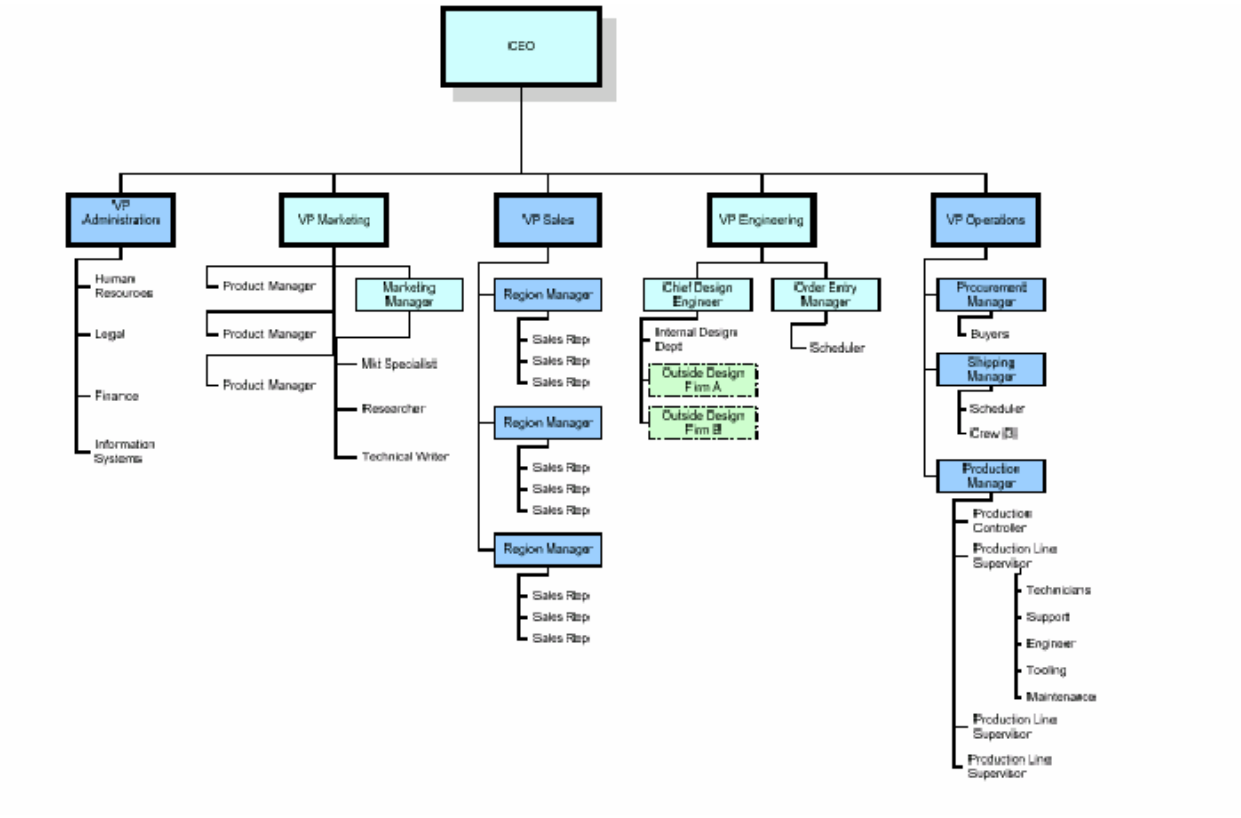

Figure 3
Processing System Hierarchy