

Today's Computing Environment

► Complexity, Complexity, Complexity

Today more than ever, IT professionals are forced to operate in a complex business environment where success is no longer determined by a single factor, such as getting the best computer system or having the right technical skills. Now, complexity is the norm — and every aspect of the environment is becoming more complex. In the following sections, we identify six key areas of complexity.

Multiple Technologies and Protocols

Not long ago, mission-critical systems all ran on mainframe technology. With the advent of the PC, local area networks (LAN) and related technologies, many business applications moved from the protected realm of the mainframe data center to the free-flowing desktops of users.

At the host or central processing area, UNIX-based midrange computers and workstations arrived, handling scientific applications more gracefully, and serving as the breeding ground for the Internet. In place

of mainframes, IT professionals can now choose specialized fault-tolerant or symmetric multiprocessing systems, or even low-end PC-based servers and workstations. Today's hottest gadgets include Palm PCs and other devices that connect to PCs and networks, allowing users to carry data in their pockets.

Once, if your system talked SNA, IBM's proprietary networking communication protocol, you could be understood by virtually all important systems and components. Today, you need to be fluent in TCP/IP and other protocols. Even your options for implementing networks can be overwhelming: Ethernet, Fast Ethernet, Gigabit Ethernet, ATM, ISDN, frame relay, xDSL, and many others.

Computing systems are no longer restricted to running on yesterday's computing platforms of choice: IBM's MVS or VSE. They may now run on UNIX, Linux, OS/400, Mac OS, Windows, OS/2, NetWare, Palm, Java-based devices, and many other alternative platforms. Even within these platform families, numerous versions and releases exist, and these are not necessarily compatible with each other. For example, UNIX has over 40 variants, even without counting the multiple distributions of Linux. Windows has Windows 3.1, Windows for Workgroups, Windows 95, Windows 98, Windows NT, Windows CE, and now Windows 2000 and Windows ME.

When dealing with the architecture or system deployment strategy, no longer are you bound to use host-based configurations where a large, powerful central computer does all the processing, and users interact via dumb terminals. In today's client-server architectures, each computing resource can be a client, a server, or both at different times. With this architecture, the mainframe is regarded as a fat server, and the dumb terminal becomes a thin client. Alternatively, you can choose a fat client (a powerful PC) that communicates with a thin server, or something in between — or a newer, Web-centered, or n-tier architecture. Each of these approaches presents unique deployment, management, and availability challenges.

Multiple Vendors

They say that the beauty of democracy is that everybody has freedom of choice. If so, nowhere is democracy practiced more fully than in the computer industry. You can buy from scores of vendors whose prod-

ucts implement the same popular technologies and standards, and safely assume that your products will work — *most* of the time, at least. Moreover, with the explosion in global electronic commerce, you can purchase software from anywhere on Earth, via the Internet, and download it directly from its authors. No longer are your choices limited to products sold by dealers and suppliers in your vicinity: you have direct access to the developers.

But freedom can be abused, and even in the best case, it creates enormous challenges for IT professionals, who are called upon to get multi-vendor products to work together with mission-critical reliability.

Varied Users

Technology is a great enabler, empowering individuals to perform equally well, wherever and whoever they are. Gone are the days when all requests for computations or data manipulation would have to be submitted to the data center. Today, nearly everyone in the organization has access to some computing resources.

Managers of every division can now access executive information systems, data warehouses, and corporate intranets for mission-critical decisions. Employees do most office-related work on PCs. Even contractual or temporary hires are provided at least limited access. And with the growth of the Internet and the proliferation of corporate web sites, customers can access the system for the latest product information or communicate with anybody in the organization via email.

As a result, IT professionals must ensure that their systems take into consideration the skills, experience, and language of a wider range of users than ever before.

Multiple Locations

With the growth of networks comes the challenge of managing computing resources that are physically distant from each other. In the 1960s, the IT organization only needed to worry about its “glass house” — the room where its giant mainframe was protected. Today, you must provide extensive remote user access. You can connect your system to the public telephone network, rent leased lines to remote

departments or offices, even connect to the Internet. Your employees want to work from their homes, or from wherever their job takes them — to another building, city, province, or country. You must somehow manage these users also.

Rapid Change

Anybody who follows the information technology industry can attest to the fact that the rate of new product developments is exponentially growing. Companies once went several months without new product announcements. Now, not only do companies introduce new products (or versions of their products) more often, but many more companies are involved. A few years back, you could read back issues of two computer magazines and still be confident that what you were reading was current technology. Today, if you read an issue that's two months old, you know that what you're reading is well on its way to obsolescence. Web technologies such as HTML and XML rarely (or barely) reach full standardization before they are updated with newer versions.

Greater Business Demands

Information technology is no longer a matter of competitive advantage: it is a matter of survival. Your customers now routinely demand what were once “extra” features and capabilities. “What’s your Web address so I can get more information about your products?” “Do you have an email address where I can send my problems or concerns?” “Can I do business with you electronically, and do away with all these paper forms?” “Can I access my bank account from the Internet?”

A Daunting Environment To Work In

To summarize, IT professionals are living in a world where they must deal with many different products from many different sources, deploying and managing them efficiently, to the satisfaction of a wide spectrum of possible users.

► The Total Cost of Ownership Issue

Many IT shops have discovered that large chunks of their costs have not been budgeted for properly. In response, consultants have come up with new Total Cost of Ownership (TCO) models for accurately costing today's distributed computing systems, and for reflecting previously unbudgeted expenses.

Total Cost of Ownership Defined

The TCO of a computing system is an organization's total cost for acquiring *and maintaining* that system.

Once, many IT professionals only factored in the costs of purchasing hardware and applications. This wasn't surprising: they grew up in the relatively easy-to-manage world of mainframes, where these were, by far, the most important costs to consider. Now, however, in the era of e-business, PCs and client-server systems, the amount of work needed to *manage and maintain* systems has become overwhelming.

End users cannot be blamed for taking it upon themselves to deploy departmental applications, when a few years ago they could not get their mainframe gurus to deliver solutions quickly enough. But now that these systems are in place, there are more components, locations, and users to manage — and the costs of computing now go far beyond the costs of acquisition.

What should go into the computation of the TCO of any system?

- **Acquisition cost** comprises the cost of acquiring the system, and includes the costs to:
 - **Research** possible products to buy
 - **Design** the system and all the necessary components to ensure that they work well together
 - **Source** the products, which means getting the best possible deal from all the possible vendors
 - **Purchase** the products — the selling price as negotiated with the chosen supplier
 - **Install** the system
 - **Develop** or customize the applications to be used

- **Train** the users
- **Deploy** the system, including transitioning existing business processes
- **Cost of maintaining availability** of the system to the end users, which covers:
 - **Systems management**, including every aspect of maintaining normal operations, such as activation and shutdown, job control, output management, backup and recovery.
 - **Maintenance of hardware and software components**, including preventive and corrective maintenance, and housekeeping.
 - **User support**, including ongoing training, help desk facilities, and problem support.
 - **Environmental factors**, a system's external requirements for proper operation, such as air conditioning, power supply, housing, and floor space.
 - **Other factors** that do not fall in any of the above categories, depending on the type of system deployed and the prevailing circumstances.

All these seem straightforward, but quantifying each cost is difficult if not impractical in today's world, because few organizations have an accounting practice that is mature enough to break down expenses in sufficient detail.

For example, we know of no organization that records all employee activities by task, information you would need to answer questions like these: *What support costs did you incur last month? How much time did each user spend in solving computer-related problems? How much work was lost due to downtime on desktop PCs?*

Additionally, companies rarely have accurate inventory and asset information regarding their computing systems, especially in large, distributed computing environments where PC, server and LAN purchasing decisions are often handled locally.

So, what's the value of knowing a system's TCO? Obviously, our objective is not to calculate exact figures. Rather, you need to understand what these costs *could reasonably be* in your organization. You must plan for these costs, even if you can only roughly estimate them. The TCO also provides

a good basis of comparison between alternative system deployment strategies, between platforms, and between competing products.

Industry TCO Estimates

When IT and user labor costs are factored in, industry consultants have estimated the TCO of typical office PC systems at roughly \$10,000 per unit, *per year*. Compare this with the typical PC system cost of around \$1,500, and you can see that the hardware and software costs account for only 15 percent of the total cost of ownership.

TCO computations for other system configurations for distributed network computing agree: hardware and software costs account for only a small portion of the total cost of ownership of *any* contemporary system. As you provide more functionality and capability to end users, TCO rises. As you install more software or provide more complex hardware, you pay increasingly more for support and maintenance.

As you can see, TCO provides a good model for evaluating computing costs — visible and invisible, budgeted and unbudgeted. We do not say that TCO should be your sole determining factor for choosing a system. However, you should be aware of these costs and plan for them. You must always balance the costs of providing a system versus the benefits to the business.

What TCO Studies Reveal

TCO studies of PCs have identified several key hidden, unbudgeted costs:

- **Fiddle factor** — Users often spend excessive time changing minor look-and-feel items on their PCs — time that could be spent performing productive work. Examples are: changing how the Windows desktop looks (e.g., color, size, icons, screensavers); installing applets or utilities (e.g., pop-up messages, animated cursors, desktop accessories); and trying out different fonts or lettering styles in documents. These activities can distract users from the more important task of ensuring quality content.

- **Peer support and self-help phenomenon** — When end users encounter problems, they rarely seek IT help. They either try to solve the problem on their own, or ask colleagues to assist, taking coworkers away from their primary job responsibilities. Not only that, as users try to gain as much computer expertise as possible, they often neglect the skills they need in their line of work. Most of their computer expertise is learned informally, by time-consuming experimentation that often causes even more complex problems.
- **User-introduced problems** — Often, users themselves cause unnecessary downtime and lost productivity through their own activities:
 - Deleting critical system files by accident or experiment
 - Changing parameters in the Windows system registry or control panel
 - Installing new software that causes incompatibilities or virus infections
 - Installing counterproductive software (e.g., games)

Problem areas in distributed client-server environments include the following:

- **Problem support** — Users need access to quality support on a timely basis. Most users revert to peer support or self-support when they cannot get the level of assistance they need. The IT organization typically fails for the following reasons:
 - It is unreachable or too distant (not necessarily physically) from users
 - It did not publicize who to contact for what types of problems
 - It does not know how to support the user's configuration (common in those systems where the users bought the systems on their own)
 - It does know how to support remote servers and distributed networks
 - It takes too long to solve user problems
 - It has no time to deal with end-user problems

- **Inventory management and control** — Often, the IT organization cannot track the configuration of most of the systems in use, especially in large organizations. When you factor in systems installed in remote locations or used by telecommuters, the situation can become even more chaotic. In order to plan for, and deliver quality user support and asset management, IT organizations must *at minimum* know the number of units installed, who owns each system, and each system's hardware and software configuration.
- **Software configuration and update management** — IT organizations are called upon to install hardware or software upgrades to systems efficiently and in a timely manner — and PC software gets updated often. There must be a way to make updates remotely, without visiting every server and desktop. There must also be a mechanism for controlling configuration files on user workstations and regulating the installation or modification of software. These requirements were never a concern back in the mainframe-centric world, as all applications were stored and run from the central host.

The Underlying Reason for High TCO

Where a company's systems have especially high TCO, its systems were most likely deployed with only the following issues in mind:

- **Functionality** — The capability of a computer to perform the tasks and run the applications required by the user.
- **Performance** — The capability of a computer to respond to user input as quickly as possible (often referred to as *system response time*).
- **Capacity** — The capability to handle growth in concurrent users, amount of data processed, number of transactions completed, or other metrics.

After the systems were deployed, issues not directly related to these criteria cropped up — issues that proved every bit as important to users over the long term. These *post-deployment requirements* include:

- **Availability** — The system or application is there when the user needs it
- **Ease of use** — No complicated procedures to learn or remember
- **Assistance** — If the user has a problem, help is easily accessible
- **Security** — The user's work is protected from loss or unauthorized access

In all cases where the TCO of a system is unnecessarily high, it is because the system or application was designed without taking into consideration the post-deployment user requirements above, particularly: availability, security, and assistance.

A Typical Scenario: Choosing Office Systems

Here's an example of how inattention to post-deployment requirements can lead to high TCO. In most large companies, the decision to purchase PCs and client-server systems is often delegated to departments, often without any guidelines from the IT organization. Users (and most young IT organizations), unaware of the need to consider post-deployment requirements, make choices based solely on functionality, performance, capacity, and price.

Given the maturity of PC and server technology and the relatively stable standards in designing LANs today, almost any hardware can qualify. As a result, inexperienced purchasers often choose unbranded systems sold at lower prices, rather than brand-name systems made by vendors such as Sun, IBM, Compaq, and Dell.

Retailers typically assemble unbranded computers using components sourced from multiple manufacturers. Unfortunately, these systems often suffer from poor quality control. Purchasers are likely to encounter unknown bugs because of substandard design or assembly; high failures due to poor quality components; poor after-sales support; and little or no system documentation. Having disregarded post-deployment requirements, they end up paying more for:

- **Support** — IT organizations must develop their own expertise since they cannot rely on the retailer, and they may have to spend more time researching problem resolution. One company spent a sizable amount simply searching for current software

patches, hardware drivers, and BIOS tuning parameters. Even so, they could not get the same level of performance as branded computers with comparable hardware and software configurations.

- **Lost user productivity** — If a system intermittently fails, or if other problems occur, system downtime can be far longer than normal.
- **Repairs** — Because warranties may not be clearly documented, and retailers often blamed problems on users, IT organizations often find themselves replacing suspicious components simply to avoid hassles.
- **System security** — Most unbranded computers have no means to prevent access inside the casing; most well designed branded computers have locks or other security mechanisms. One company we encountered had problems with missing memory chips they suspected were being stolen by employees for use in home computers.

Availability as the Most Significant Contributor to TCO

Our experience with information systems has shown us that the user requirement responsible for the greatest hidden costs is *availability*. This user requirement takes precedence over all others. What good is a system if it is unavailable? It also requires ongoing management and maintenance throughout the entire life of every system.

A system is available when users can work with it without experiencing outages. Note that for as long as the user does not perceive or feel the outage, the system is available to him. Availability is measured from the user's point of view. A user will consider a system unavailable if:

- **The system is not accessible** — If the users cannot access the resources they need to run their application, the system is unavailable. The system is equally unavailable if all workstations or software licenses are in use, or if the network connection to necessary data is down, or if the system has a virus infection.

- **The system is running too slow** — The system may be operational but if the response time is long, the user will give up waiting and consider the system as unavailable.
- **The system is intermittently having problems** — The user will choose not to use a system if she suspects her work may be lost due to intermittent system failures.

► Summary

In today's client-server-dominated IT environment, we must understand TCO in order to effectively evaluate all of our deployment alternatives. All studies on TCO have shown that the TCO of interconnected servers and workstations is high compared to the centralized mainframe and dumb terminals of yesteryear, and the key reason is inattention to post-deployment requirements, especially availability.

Availability deals not only with the prevention of “real” system outages, but with user-perceived outages as well. These perceived outages are anything that prevents the user from working with the system productively, such as prolonged response times, lack of assistance, or lack of available workstations.

We can slash TCO by designing systems and applications with availability in mind. In the next section, we will review availability requirements in greater detail, in order to properly address them.