

# Total Cost of Ownership: Principles and Practical Applications



By [Floyd Piedad](#)

Date: Dec 6, 2001

Article is provided courtesy of [Prentice Hall Professional](#).

[Return to the article](#)

---

"Cheap" computer technology ain't necessarily cheap. When evaluating what something will cost to purchase, are you giving any thought to the total cost of owning it? What will it cost you to use it? Maintain it? Repair it? Upgrade it? Will end users adopt it? Will they spend all of their valuable working time playing with it? Fighting with it? Complaining about it?

---

The total cost of ownership (TCO) of a computing system is defined as the total cost for acquiring, activating, and *keeping* that system running. It's an accounting methodology that today is proving to be crucial in making sound IT decisions.

Many IT professionals conveniently factor in only the costs of purchasing hardware and software when doing TCO analysis. This isn't surprising; when pressed for time, they only take into account what's easy to find out. In the relatively easy-to-manage world of mainframes and big centralized computing systems, hardware and software accounted for much of the cost factors. In the current era of e-business, client/server, and peer-to-peer systems, however, the costs of *managing and maintaining* computer systems is often much higher and cannot be ignored.

What should go into the computation of the TCO of any system? We can group these costs into *direct* and *indirect* costs.

- **Direct costs** pertain to the acquisition expenses or the cost of buying the system, and cover all of the following activities:
  - **Researching** possible products to buy, which is essentially a labor cost but may also include materials cost, such as purchase of third-party research reports or consultant fees.
  - **Designing** the system and all the necessary components to ensure that they work well together. Naturally, this cost component will be higher if a move to a totally different system platform is being considered.
  - **Sourcing** the products, which means getting the best possible deal from all possible vendors through solicited bids or market research. It's often sufficient to get a quotation from three vendors (with the cheapest one not necessarily being the best choice). With the Internet, it's even easy to get price quotations from sources outside the country, to get a good spectrum of pricing options.
  - **Purchasing** the product(s), which includes the selling price of the hardware, software, and other materials as negotiated with the chosen suppliers. Include all applicable taxes that might be incurred. Don't forget to consider the costs of the systems at the end-user side; some system choices might entail a change or upgrade at that end.
  - **Delivering** the system, which includes any shipping or transportation charges that might be incurred to get the product into its final installation location.
  - **Installing** the system. Bear in mind that installation also incurs costs in utilities and other environmental—*not* just labor costs. If the installation of the system will result in downtime for an existing system, relevant outage costs must be included. Any lost end-user productivity hours during this activity should also be factored in.
  - **Developing or customizing** the application(s) to be used.
  - **Training** users on the new system.
  - **Deploying** the system, including transitioning existing business processes and complete integration with other existing computing resources and applications. Include here the costs to promote the use of the new system among end users.

- **Indirect costs** address the issues of maintaining availability of the system to end users and keeping the system running, which includes the following:
  - **Operations management**, including every aspect of maintaining normal operations, such as activation and shutdown, job control, output management, and backup and recovery.
  - **Systems management**, such as problem management, change management, performance management, and other areas.
  - **Maintenance of hardware and software components**, including preventive maintenance, corrective maintenance, and general housekeeping.
  - **Ongoing license fees**, especially for software and applications.
  - **Upgrade costs** over time that may be required.
  - **User support**, including ongoing training, help desk facilities, and problem-resolution costs. Remember to include any costs to get assistance from third-parties, such as maintenance agreements and other service subscriptions.
  - **Environmental factors** affecting the system's external requirements for proper operation, such as air conditioning, power supply, housing, and floor space.
  - **Other factors** that don't fall into any of the above categories, depending on the type of system deployed and the prevailing circumstances.

All these cost factors seem fairly obvious, but quantifying each cost is difficult or impractical in today's world, because few organizations have an accounting practice that's mature enough to identify and break down all these types of expenses in sufficient detail. For example, very few organizations record all employee activities by task and hours used—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, multi-location computing environments where PC, server, and local network purchasing decisions are often handled at the department level.

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. A fair amount of intelligent "guesstimation" is much better than blindly deciding on an IT solution on the basis of sticker price alone. In addition, TCO analysis provides a good basis of comparison between alternative system-deployment strategies, between platform choices, 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 from as low as \$3,000 to as high as \$10,000 per unit, *per year*. Note that typical PC hardware and software prices range from a low of \$700 to a high of \$2,000 for desktop units.

An example of how TCO can help in making a decision on system migration is a recent analysis by the Gartner Group that estimates the migration costs per PC system going from a Windows 98 to a Windows 2000 platform to be anywhere from \$2,000 to \$3,000. The same sort of analysis by Giga Group—but quantifying the labor savings gained—puts the cost of migration at \$973 per system. In Giga Group's approach, they tried to quantify the gain in user productivity hours from the use of the much more stable Windows 2000 operating system.

Although all analyst's TCO estimates vary considerably, they all point to the fact that

- TCO results will be very different for every organization, given their varied computing environment, user experience level, and IT expertise.
- PC systems have much higher indirect costs than direct costs.
- TCO analysis is never going to be an exact science, due to the many assumptions and unknowns that have to be taken into account.
- As you provide more functionality and capability to end users, TCO rises. As you install more software or provide more complex hardware at the hands of end users, you pay increasingly more for support and

maintenance.

TCO provides a good model for evaluating computing costs—direct and indirect, visible and invisible, budgeted and unbudgeted. Of course, TCO cannot be your sole determining factor for choosing a computing system. What I'm driving at here is that you should be aware of these costs and plan for them.

At the same time however, you must always balance the costs of providing a system versus the benefits it brings to the business and the end users. Many decisions you make will not be due to cost-avoidance but rather on the basis of business advantage. Case in point is having Internet connectivity. On one hand, providing such a facility for the enterprise means additional investments in firewalls and other security products, as well as a dramatic rise in potential damage from hackers, viruses, and other malicious activities. But on the other hand, what business can adequately compete or even survive without the access to information, worldwide reach, and accessibility to customers that the Internet provides?

## What TCO Studies Reveal

TCO studies of PCs, PDAs, and other end-user-oriented computing platforms have identified several key, hidden, and oftentimes un-budgeted costs due to the following phenomena:

- **Fiddle factor.** Users often spend excessive time changing minor look-and-feel items on their systems—time that could instead be spent performing productive work. Examples include changing how the Windows desktop looks (color, size, icons, screensavers), installing applets or utilities (pop-up messages, animated cursors, desktop accessories), and trying out different fonts or lettering styles in documents. These activities distract users from the more important task of ensuring quality content in their work.
- **Peer support and self-help phenomena.** 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 themselves and their coworkers away from 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 skill 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 activities such as these:
  - Deleting critical system files by accident or experimentation.
  - Changing parameters in the Windows system registry, control panel, and other configuration files.
  - Installing new software that causes system instabilities, security exposures, or counterproductive activities (for example, utilities or games).

## 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 objectives 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's because the system or application was designed without taking into consideration the post-deployment user requirements above, particularly availability, security, and assistance.

## Availability as the Most Significant Contributor to TCO

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's unavailable? Availability also requires ongoing management and maintenance throughout the entire life of every system.

A system is considered available when users can work with it without experiencing outages. Availability is measured from the user's point of view. It 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. As long as the user doesn't perceive or feel the outage, the system is considered "available."

A user will consider a system unavailable if one of these conditions occurs:

- **The system is not accessible.** If the user can't access the resources he or she needs to run an application, the system is considered 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 slowly.** 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 he or she suspects that work may be lost due to intermittent system failures.

## TCO Summary

In today's widely distributed IT computing 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, workstations, and intelligent access devices is higher compared to the centralized mainframe and dumb terminals of yesteryear, and the key reason is the lack of attention to post-deployment system requirements, most especially the availability requirement.

If a system is designed, deployed, and managed without special attention to organization, people, process, and technology issues, the total cost of ownership will definitely spiral out of control.

## Harris Kern's Enterprise Computing Institute

The underlying objective of [Harris Kern's Enterprise Computing Institute](#) is to be the world's foremost source of information on managing enterprise computing in the 21st century. Organizations that master our approach and techniques will ensure that their IT infrastructure—comprising mainframe, client/server, and Internet computing—are closely aligned with their business objectives. Surprisingly, technology is the easy part. The key is in taking a comprehensive approach that includes people, process, and organization.

Our approach is well-documented in our books, white papers, and articles. Like the James Martin books that provided guidance to mainframe computing in the early years and the more recent *Dummy* series for PC computing, The Enterprise Computing Institute series of books provides guidance to enterprise computing for all IT professionals in the 21st century.

Our approach and techniques are proven. Placing special emphasis on organization, people, process, and technology, we have assisted numerous companies to assess, build, and manage complex distributed infrastructures. The companies include, among others, The Royal Hong Kong Jockey Club, Standard & Poors, Twentieth Century Fox, U.S. Satellite Broadcasting, The Weather Channel, several Time Warner companies, Warner Brothers, CNF, TransAmerica Corporation, Sony Pictures, Fannie Mae, and Hong Kong International Terminals. Over the next few months we will describe our approach, methodologies, processes, organizational structures, and technology architectures in a series of exciting articles—articles that will allow you to build a world-class IT infrastructure.