

Estimating the real total cost of ownership

By Harris Kern

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 is an accounting methodology that is proving to be crucial in making sound IT decisions in today's marketplace. Here is a detailed examination of exactly what needs to go into the computation of any system's TCO, from the direct costs, which are generally well known, to the indirect costs, which are sometimes ignored in the calculation.

Direct costs

Direct costs pertain to the acquisition expenses or the cost of buying the system. These costs are incurred when:

- **Researching** possible products to buy. This is essentially a labor cost, but it may also include materials costs, such as purchases 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 the possible vendors through solicited bids or market research. It is often sufficient to get a quote from three vendors (with the cheapest one not necessarily being the best choice). With the Internet, it is now very easy to get price quotes from sources outside the country in order to get a good spectrum of pricing options.
- **Purchasing** the products. This cost is made up of the selling price of the hardware, software, and other materials as negotiated with the chosen suppliers. Include in this amount any and all applicable taxes that might be incurred. Do not forget to consider the costs of the systems on the end-user side, as some system choices might entail a change or upgrade at their end.
- **Delivering** the system. This cost is any shipping or transportation charges that might be incurred in getting the products to their final installation locations.
- **Installing** the system. This is often difficult to calculate, but bear in mind that installation includes more than just labor costs—it also includes items such as costs in utilities and other environmental factors. If the installation of the system will result in the downtime of an existing system, then the outage costs must be included as well. Any lost end-user productivity hours during this activity should also be factored in.
- **Developing** or customizing the applications to be used.
- **Training** the users on using the new system.
- **Deploying** the system. These costs include transitioning existing business processes and the complete integration with other existing computing resources and applications. Also included are the costs to promote the use of the new system among the end users.

Indirect costs

Indirect costs address the costs of maintaining availability of the system to the end users or keeping the system running. These costs cover:

- **Operations management**, including every aspect of maintaining normal operations, such as activation and shutdown, job control, output management, backup, and recovery.
- **Systems management**, such as problem management, change management, performance management, and others.
- **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. Don't forget to include any costs to get assistance from third parties, such as maintenance agreements and other service subscriptions.
- **Environmental factors**, which cover a system's external requirements for proper operation, such as air conditioning, power supply, housing, and floor space.
- **Other factors** that do not fall into any of the above categories, depending on the type of system deployed and the prevailing circumstances.

The numbers don't have to be exact to be valuable

All these cost factors seem fairly obvious, 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 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, which is the type of information you would need to answer questions like:

- What support costs did you incur last month?
- How much time did each user spend 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, multilocation computing environments where PC, server, and local networking purchasing decisions are often handled at the department level.

The objective in knowing a TCO is not to calculate exact figures. Rather, you need to understand what these costs could reasonably be for your organization. You must plan for these costs, even if you can only roughly estimate them. A fair amount of intelligent "guestimating" 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 for comparison between alternative system deployment strategies, platform choices, and 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 [Gartner](#) 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. [Giga Information Group](#) performed the same sort of analysis but, by quantifying the labor savings gained, the cost of migration was set at \$973 per system. In Giga's approach, it tried to quantify the gain in user productivity hours from the use of the much more stable Windows 2000 operating system.

Although all analyst TCO estimates vary considerably, there is some consensus that:

- TCO results will be very different for each 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. While it is not true that TCO should be your sole determining factor for choosing a computing system, it is important that you are aware of these costs and plan for them.

At the same time, however, you must always balance the costs of providing a system with the benefits the system brings to the business and the end users. Many decisions you make will not be due to cost avoidance but will be made on the basis of business advantage. A 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, the worldwide-reach, and the 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 unbudgeted 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 are: changing how the Windows desktop looks (e.g., color, size, icons, screen savers); 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 distract users from the more important task of ensuring quality content in their work.

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 them and their coworkers away from their primary job responsibilities. Not only that, but 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 skills are 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, such as:

- 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 (e.g., utilities, games).

The underlying reason for high TCO

When a company's systems have especially high TCOs, 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 has to be there when the users need it.
- **Ease of use:** There should be no complicated procedures to learn or remember.
- **Assistance:** If the user has a problem, help needs to be easily accessible.
- **Security:** The user's work needs to be 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.

Availability as the most significant contributor to TCO

Availability is almost certainly the user requirement that is responsible for the greatest hidden costs. This user requirement takes precedence over all others, because, of course, what good is a system if it is unavailable? Availability 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 or her. Availability is measured from the user's point of view. 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.

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, 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 unavailable.
- **The system is intermittently having problems:** Users will choose not to use a system if they suspect their work may be lost due to intermittent system failures.

TCO summary

In today's widely distributed IT computing environment, you must understand TCO in order to effectively evaluate all of your deployment alternatives. Studies on TCO have shown that the TCO of interconnected servers, workstations, and intelligent access devices is higher when compared to the centralized mainframe and dumb terminals of yesteryear. The key reason for this is the lack of attention to post-deployment system requirements, especially the availability requirement.

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

For more information, visit the [Harris Kern Enterprise Computing Institute](#) Web site.