Calculating Total Cost of Ownership when Choosing a Solution

~ By Curt Finch

Total cost of ownership (TCO) is a financial estimate that helps consumers and enterprise managers determine direct and indirect costs of a product or system. TCO goes beyond the initial purchase price or implementation cost to consider the full cost of an asset over its useful life. A TCO analysis often shows there can be a large difference between the price of something and its long term cost.

For example, let’s say you buy a car that’s inexpensive, but it breaks down constantly. When you take it to get fixed, you find that the repair shop is far away and the parts are costly. It also loses value faster than other cars that cost more when it comes to resale. Your time is also valuable and all of the trips to the repair shop should be taken into consideration, too.

So, let’s do the math:

- **Car A**
  - Initial cost: $10,000
  - 5 years of repairs, plus normal maintenance: $7,500
  - Value after 5 years when you sell it: $1,000

- **Car B**
  - Initial cost: $20,000
  - 5 years of repairs, plus normal maintenance: $2,000
  - Value after 5 years when you sell it: $10,000

The 5 year TCO of Car A is $16,500, whereas the 5 year TCO of Car B is $12,000. Even though Car B was more expensive up front, it costs less in the long run.

How TCO is Calculated and Used to Make Decisions

My company uses a customer relationship management (CRM) tool that tracks leads, sales and customer support issues. It’s been used in the company for a very long time and is deeply embedded in our company’s operations. However, the usability and performance are quite poor, and the monthly cost is exorbitant. There are plenty of competitors to this CRM system and the time has come to consider other options.
In order to make this decision, the TCO of each of the potential solutions must be understood. The future costs need to be considered for a set period of time. Let’s consider four options:

- Option 1: Stick with the existing vendor and stop whining
- Option 2: Build a solution internally, perhaps using open source tools (e.g. SugarCRM) embedded in the solution
- Option 3: Choose a huge mega-vendor like Microsoft
- Option 4: Choose a small vendor that will pay closer attention to our needs

This scenario presumes that these options exist and are somewhat rational. Since this is purely hypothetical, let’s pretend they do.

**How would these solutions compare over a 5 year timeframe?**

For option 1, the cost of the monthly fees is known and we can estimate how the fees might increase over the next few years. Does that mean we understand our costs? Not really. An estimate of the opportunity cost is needed - this includes all of the things the solutions doesn’t do that we wish it did. Additionally, any company has the potential risk of merger, hostile takeover, sale of the company, division, shutting down or alteration of the availability of the technology in some way that may adversely affect their customers’ operations.

Each of these options has different costs for hardware, backups and operating system software. Software installed by your IT staff in your own company’s environment will have all of these costs. On the other hand, solutions provided directly to your browser from the vendor’s system (Software as a Service, also known as SaaS or cloud solutions) usually only have a monthly fee and any web browser will work with the software.

Vendor risk is another factor. Will your chosen technology vendor still be supporting the product in a few years? Large vendors obsolete technology platforms all the time, and this is always a risk. Assign a dollar value to this risk - it is part of the TCO for each vendor. If a system is built internally, the expert who understands the software better than anyone else is employed by your company. What if that person leaves?

Let’s take backup and disaster recovery, for example. In option 1, there isn’t as much worry about disaster recovery because the vendor offers the CRM system on a SaaS basis. Customers don’t have to conduct backups because they trust the vendor to do this. Most SaaS vendors are pretty good about doing this, and their SAS70 certification should explain the procedures they follow to the customer.

Option 3 - the large vendor - has a high cost of backup. Most companies overestimate their disaster recovery capabilities. Disk drives go bad all the time. Their "mean time between failure" (MTBF) is typically just a few years. It’s surprising how bad some IT shops are at this difficult, but critical process.

Option 2 - build a solution internally - has all the backup challenges of option 3, plus the necessity of backing up the source code modifications that have been constructed internally, test suites that have been built, and anything else associated with building and maintaining an in-house software system.

**Be Aware of Vendors’ Hidden Agendas**

All managers need to have some system of TCO in place when making decisions on choosing solutions. Vendors will try to manipulate and obfuscate the true TCO of their solution and it will be different for each installation. Customers must be sure that THEY own the definition and calculation of TCO and don’t allow the vendor to drive the agenda.

For example, some of the costs, like risk and opportunity, are nebulous and hard to define. Vendors will try to make other vendors’ solutions look risky. Large companies like to say that dealing with small companies is riskier, but this is not necessarily the case. Large firms can evaporate overnight (think Wachovia, Lehman
Brothers). Additionally, large vendors may not view you as large enough of a client to matter if they ignore your needs. This scenario sheds a positive light on option 4 - choosing a small vendor.

Opportunity cost (e.g. if we build this ourselves, will we be gaining new opportunities in the marketplace from doing this or will it distract us?) needs to be assigned a dollar value, which is difficult to do. But customers must come up with some estimate of opportunity cost for each solution being considered.

In the long run, you're the boss. It's all up to you. You get to define the costs and make sure they are a good representation of the issues your business faces. Don’t let a vendor’s salesperson take over the process from you. But don't throw the baby out with the bathwater - vendors do know the weaknesses of their competitors and can be very useful in this regard.

As you can see, each solution will have different benefits, risks and costs. Only when a thorough analysis is conducted can you compare the TCO across possible solutions.

**Putting the TCO Calculation into Practice**

Consider this list of costs and risks that are frequently overlooked and could differ substantially for the 4 options described above. Obviously you can't do this analysis in a time crunch for more than a very few possible solutions.

- **Software Cost**
  - License cost + base/server cost
  - Client side cost (if any)
  - Integration
  - Purchasing process (how many vendors?)
  - Migration

- **Operating Costs**
  - Training
  - Insurance
  - IT staff
  - Management time
  - Electricity
  - Floor space
  - Outage costs
  - Back-up/recovery cost

- **Annual cost**
  - Maintenance

- **Server cost**

- **Other product costs that are required, like a database, web-server, etc. (Even if you already have these things, you might need more, or to keep or support them longer)**

- **Risk cost**

- **Opportunity cost**
Happy planning!

Curt Finch is the CEO of Journyx http://journyx.com, a provider of web-based time tracking, project accounting and resource management software designed to guide customers to per-person, per-project profitability. Curt earned a B.S. in Computer Science from Virginia Tech in 1987 and he has been creating software or managing software teams ever since. An avid speaker and author, Finch’s book, ”All Your Money Won’t Another Minute Buy: Valuing Time as a Business Resource” is available in most online bookstores. He is also a blogger for Inc. http://www.inc.com/technology. Curt can be contacted at curt@journyx.com (mailto:curt@journyx.com)