TOA TECHNOLOGIES

Predictive

Let SaaS Put You in the Driver's Seat

Comparing the Total Cost of Ownership of SaaS with traditional on-premises solutions

A Business White Paper

US: TOA Technologies, Inc. | 3333 Richmond Road | Suite 420 | Beachwood OH 44122 | www.toatech.com

TOA TECHNOLOGIES

Contents

A Business White Paper

US: TOA Technologies, Inc. | 3333 Richmond Road | Suite 420 | Beachwood OH 44122 | www.toatech.com

*leasingbuying***8X the documented cost of the initial system implementation**[1]*salesman*

Here we have a typical automobile purchasing process flow:

1. Undiscounted 10-year life cycle. Source: The Role of Linux in Reducing the Cost of Enterprise Computing – IDC White Paper Sponsored by Red Hat, 2001

...Is like traditional software implementation

The shrewd reader will by now recognize the analogy being drawn between traditional software deployment and building, rather than buying, a car so let us not belabor the point. Traditional on-premises software implementations carry risks and expenditures that are considerable and often overlooked when initially calculating the costs. Parts from various software and hardware suppliers must be assembled, integrated and maintained, with the ultimate responsibility and cost falling on the buyer. Businesses deploying package software assume considerably more responsibility and downstream costs than the metaphorical car buyer.

In his book, "Software Project Secrets: Why Software Projects Fail", author and system developer George Stepanek quotes a finding by the Standish Group that only 28% of software projects during the study period (2001) were considered successful, of the rest, 63% were substantially late, 45% over budget and one-third were lacking in features[2] (the percentages add up to greater than 100% because some projects had multiple reasons for failure). Many businesses have turned to SaaS (or "Cloud-based" solutions) in order to mitigate these risks and to improve the financial return from new software deployments.

Not so much hidden as overlooked

There is really no such thing as a "hidden" cost, all costs find their way to the bottom line where they are perfectly obvious as part of the organization's P&L. There are however, unanticipated, unidentified and unmeasured costs which, if they remain so will become unmanageable costs. The rigor of TCO is to identify these sometimes hard to measure costs so that they can be better managed and support better decision making.

Any reasonable comparison between the cost of a SaaS and on-premises solution will of course include the basic elements of hardware and software that are required to be purchased such as:

Hardware

- **Application Servers**
- **Database Servers**
- **Redundant Application Servers**
- **Redundant Database Servers**
- **Hardware Support**
- **Shared or dedicated field or client equipment**

Software

- **Application License**
- **Database License**
- **Operating System Licenses**
- **Licenses for associated tools and supporting applications**
- **Software Maintenance and Support Agreements**
- **Training**
- **Help Desk**

2. "Software Project Secrets: Why Software Projects Fail" - George Stepanek, Apress 2005

These items are generally identifiable as the direct costs of software ownership and represent, by most accounts, *about one third of the cost of owning and operating the software*. The other two-thirds of the cost come as indirect costs.

In their book on high availability system design, Floyd Piedad and Michael Hawkins divide TCO into two aspects, acquisition costs and costs to maintain high availability . Acquisition costs are similar to the items described above, adding items like research and design. These costs are relatively easy to identify, quantify and enter into the financial analysis master spreadsheet. But the authors go on to describe the significant costs involved in maintaining availability of these system for end-users which they characterize as harder for businesses to quantify but critical to understanding TCO.

Piedad goes on to assert that on-premises systems that ultimately result in high TCO tend to be ones that were initially evaluated only on the basis of functionality, performance and capacity and where cost issues not related to these criteria appeared only after deployment. Of course hindsight is always 20/20 but many of the costs of maintaining a system's availability can be anticipated and incorporated into the pre-purchase analysis.

Let's use these High Availability Maintenance categories as our framework for examining the "hidden" or overlooked costs of on-premises solutions.

Cost Considerations for Maintaining High Availability Systems

- **Systems management,** "including every aspect of maintaining normal operations"
- **Maintenance** of hardware and software components
- **User support,** this includes problem management and ongoing training
- **Environmental factors** such as air conditioning, electricity and floor space
- **Other factors**

Maintaining Availability

Wouldn't it be great if once business systems were bought and implemented they just worked tirelessly, 24x7, without downtime and without maintenance? If they did we wouldn't need help desks, consultants or maybe even IT departments. But of course organizations with Mobile Workforces know all too well that even well-designed systems fail from time-to-time and that maintaining availability to end-users is a demanding task (or we wouldn't need Mobile Workforces either).

Systems Management

Many purchase-time analyses don't fully account for the resources that are required to run the system. The components of Systems Management are organization, procedures and tools. Some examples of areas where effort and costs are often unaccounted for are:

- Escalation procedures, identification, documentation, testing
- Training and integration of help desk staff
- Tools and procedures for identifying, documenting, reporting, correcting and testing post-implementation software and hardware incidents
- Troubleshooting procedures development, processes to identify whether problems that occur are at the application, platform (e.g., database), interface or hardware level

All of the costs to develop, manage and review these items will be assumed by the business implementing on-premises software. By contrast, many of these requirements, and their associated costs, are the responsibility of the SaaS provider in a SaaS implementation.

3. High Availability: Design, Techniques and Processes – Floyd Piedad and Michael Hawkins, Prentice Hall 2001

System Maintenance

Ideally, system maintenance should be transparent to the end-user, performed overnight or over a weekend when systems aren't in use. Alas, experience tells us that even businesses that run on an 8x5 schedule find system maintenance tasks overrunning the workday. For extended hours operations like cable and utility services, this sort of problem is even more common. Operating systems, databases and layered interfaces often require bug fix, support and security patches. Sometimes applying those patches at one layer of the system results in unexpected problems at another layer.

Users of enterprise-class SaaS applications find that these issues (and their associated cost in effort and productivity loss) are typically transparent to the business. For example, in 2009 TOA Technologies' Mobile Workforce application ETAdirect had an availability of 99.995%, a reliability level that would be difficult *and expensive* for many individual businesses to attain with on-premises software. A proper TCO analysis will need to account either for the cost of maintaining such high availability or the loss of productivity resulting from a less available system.

Estimating Downtime Costs

For the purpose of calculation, downtime costs can be estimated based on Service Level Agreements (SLAs) normally developed in cooperation with IT and the system suppliers.

System	Commitment made by	Availability %	Downtime / month
Hardware availability commitment	Either the hardware supplier or IT on their behalf	99.9%	43.2 minutes
Application & Database availability commitment	Application supplier, database supplier	99.5%	216 minutes
Network availability commitment	IT	99.99%	4.32 minutes
Overall		**99.39%**	**263.52 minutes (4.39 hours)**

Once the overall availability is calculated, this figure can be used to calculate the expected impact on end-users and any resulting productivity issues. It is assumed here that the availability commitment does not include regular planned maintenance downtime.

User Support

There are some hard-to-measure differences in end-user support when comparing SaaS to traditional on-premises software. For example, with an on-premises solution using a client-server architecture, a problem reported by the end-user could be related to the configuration or operation of the client sitting on the user's desk, the problem might be occurring in the application server, it could be the database hardware or software or in any of the components in between. IT help desks will need to train and maintain the skills to be able to work with end-users to isolate the problem and to fix it, possibly in consultation with the various vendors of the software and/or hardware. Maintaining such skill sets is costly and needs to be considered as part of the cost consideration for traditional on-premises solutions.

With SaaS solutions, the complexity of the hardware and software involved in the application are the responsibility of the SaaS supplier. In this case, the SaaS vendor maintains the expertise on both the software and supporting hardware, offering end-users and IT departments alike a simpler and less-costly troubleshooting process and a reduced need to maintain application-specific integration expertise.

Environmental Factors

The cost to power and cool application and database servers is often not explicitly identified in the cost analysis for new systems. Because the business saves on these expenses when they choose a Cloud-based solution, these environmental factors for on-premises solutions come into focus.

Number of Servers:	2
Power Consumption per Server:	638 watts
Air Conditioning Requirement per Server:	638 watts
Hours in Operation per Year (24x7):	8760
Total Electrical Consumption per Year:	11,178 kWh
	$0.10
Retail Electricity Cost / kWh UK:	
Annual Server Electricity Expense U.S.:	$1,142.00
	$2,078.00

Other Factors

One key cost consideration that we have not covered to this point is Disaster Recovery and business continuity planning. There is a fundamental costing difference between Disaster Recovery for business-critical systems that are on-premises and those that are provided using the SaaS model.

Traditional on-premises solutions require that a Disaster Recovery plan be created for the system including the hardware, software and physical facilities. Issues of data concurrency, fail-over procedures, etc. must be considered and implemented. The design, planning and efforts that go into such a plan have real costs to the business even if the system under consideration is part of a larger plan. Most Disaster Recovery plans provide for off-site back-up, storage and/or archiving of data and transactions. The ongoing costs of these procedures should be included in a TCO analysis.

For businesses considering a SaaS solution the primary effort and cost with respect to Disaster Recovery for the application[4] is the due diligence in selecting the SaaS vendor. An enterprise-class SaaS supplier should be able to provide the customer with a thorough and convincing plan for continuity of their operation in the event of fire, flood or other calamity.

4. Disaster Recovery comprises a broad set of business rules covering many aspects of the businesses operations. Here we are speaking specifically about one aspect of that discipline.

Upgrades

In comparing the TCO of SaaS applications and on-premises solutions, management will need to consider the time-frame over which to perform the economic analysis. Forrester Research suggests a 10-year analysis horizon with the assumption that a software upgrade occurs in year eight[5]. One might imagine that such a long time-frame would unduly favor an on-premises solution but it turns out that in general just the opposite is true. In fact, a 10-year TCO analysis can be expected to show the Cloud-based solution as the less expensive alternative.

Forced Upgrades and Voluntary Upgrades

For the pure purposes of calculating TCO, the *reason* for an upgrade is immaterial, costs are costs. But for the purpose of clarity, we should consider two types of upgrades, forced and voluntary. Forced upgrades are ones that are driven by software and hardware vendors who require a particular installation of software to be upgraded to a new version because the vendor is withdrawing support for the version currently in use. On the other hand, a voluntary upgrade is one that is initiated by the users or the business in order to obtain a higher level of functionality, support more users or more transactions or improve operations.

Over a 10-year span, users can expect to upgrade their core database software at least once, and possibly twice. The direct cost of such an upgrade will depend on the contract the organization has with the database vendor or provider, obviously this cost should be factored into the TCO analysis. The indirect cost of such an upgrade needs to be considered as well.

At implementation-time, the application software provider will specify a set of foundational software and hardware on which their application is meant to run. Normally, this specification will include the release version down to the dot-level (and sometimes much further). In turn, the database and foundational software will be specified to run on a particular set of hardware with a particular operating system version and, often, a specific set of patches. When any one of these key components of the system is changed, it forces a re-evaluation of the entire layered architecture to ensure that all the pieces still work together properly and will be supported by the application provider. The anticipated coordination of these changes and the User Acceptance Testing (UAT) of the upgraded system are costs that must be factored into the TCO analysis.

A voluntary upgrade carries with it a similar set of capital costs, licenses, new hardware, etc., as well as operational costs, specification, testing, implementation, data cleansing/migration, etc. According to Forrester, organizations should budget about 65% of the original implementation costs for such a downstream upgrade.

By contrast, SaaS implementations engender neither of these upgrade costs. Changes to foundational software are handled by the SaaS provider and should be transparent to the end-user. Moreover, SaaS users benefit from a continuing set of upgrades that are made available to customers, avoiding the Big-Bang approach often necessary with traditional on-premise upgrades.

5. Comparing The ROI Of SaaS Versus On-Premise Using Forrester's TEI™ Approach – R Ray Wang, 2006

Repairing the Analogy

We began by arguing that the car analogy, lease versus buy, was not a correct one for comparing SaaS to traditional on-premises solutions, now let us return to the analogy and see if it can be corrected to make it more accurate.

We devoted a section above to building a kit car, drawing comparisons to implementation risk and costs to maintain a high-availability business-critical system in terms of skills and effort. To that, we added the fact that upgrades will be required to foundational software during the application's lifetime, resulting in costs for re-integration and re-testing of the system.

So our car analogy will have to include the notion that key components of our purchased car, say the transmission, will need to be replaced occasionally with new components that were not part of the original specification. These new components may or may not have the same dimensions and fittings as their predecessors. The parts themselves may or may not be under warranty, but the cost of integrating them into the car will be born by the car owner.

We are more satisfied with the comparison of an enterprise-class SaaS application to leasing a car, with one exception. For our TCO analysis of a traditional on-premises system, we included costs for upgrades. Presumably, at least in the case of voluntary ones, those upgrades are meant to improve some aspect of the end-user experience. Upgrades are normally part-and-parcel of the SaaS value proposition, with users being able to take advantage of a series of upgrades that are regularly introduced to the Cloud-based product. Therefore, our car lease contract will need to stipulate that whenever a new model of our leased car is introduced, the manufacturer delivers it to us in trade for ours. An excellent leasing model indeed!

With these changes, we have for the most part but this analogy right and, most likely, stretched it to its very limit.

Conclusion

In this paper we have tried to demonstrate that to make an apples-to-apples financial comparison between similarly tasked SaaS applications and traditional on-premises applications, a Total Cost of Ownership analysis is required. We reasoned that many of the management tasks required to run an on-premises solution, the costs for which will most assuredly be born by the business, do not accrue with a Cloud-based software (or, if you prefer, are already "cooked into" the pricing model). Moreover, we asserted, with evidence, that these costs are considerable and are likely to be *multiple-times larger than the initial implementation costs* for the on-premises hardware and software. Finally, we spoke of risk, implementation and operational, which is probably the most difficult of concepts to quantify but should be considered as a financial differentiator between the two classes of solutions.

There is no doubt that there will be cases where an on-premises solution makes financial sense. But most businesses will find it unacceptably expensive to develop the core-competencies in the application hardware and software that match those offered by Cloud-based software solutions such as TOA Technologies. Evidence suggests that taking advantage of the scale, operational and application expertise of an enterprise-class SaaS provider such as TOA Technologies, frees up physical, financial and human resources that can be re-tasked to more productive core-business activities.

About TOA Technologies

TOA Technologies is the leading provider of Cloud-based mobile workforce management applications for large enterprises. Its patented platform improves customer service while dramatically reducing operational costs and providing immediate return on investment. As the industry's only solution using predictive, time-based analytics, TOA reduces customer wait times while increasing field workforce efficiency. TOA automates Time Of Arrival communications across multiple channels to proactively keep customers apprised of their appointment status. It delivers unparalleled cost savings by providing on-demand tools for real-time planning, routing, dispatching and tracking of workers in the field. TOA's system deploys quickly, is highly configurable and is easily integrated with existing CRM solutions. Named a Visionary for three years running in Gartner's annual Magic Quadrant for Field Service Management, TOA Technologies is headquartered in the United States and has offices across Europe. For additional information please visit www.toatech.com.

For more information, please contact us

TOA Technologies - US
TOA Technologies - US
TOA Technologies, Inc.
3333 Richmond Road
Suite 420
Beachwood OH 44122
Tel: +1.216.360.8106
Email: marketing@toatech.com