WHITE PAPER

The Enterprise Application Lifecycle: It's Costing You More Than You Know

By Nick Cavalancia

FLEXERA® SOFTWARE

TABLE OF CONTENTS

FLEXERA® SOFTWARE

There's much more to managing applications (and the associated cost) than just pushing out a new version.

Application management was once little more than standardizing on a version of, say, Adobe Professional, which you deployed every few years. But as applications seek to stay competitive in the face of security concerns and the availability of SaaS applications, the speed at which application updates are released have never been faster. And without proper control over what is installed where, that can cause application conflicts, wasted licensing costs, and potentially soaring support costs.

So how can you manage your applications and minimize your total cost of application ownership?

While most of you employ at least a basic form of automation to take the burden of implementation off your plate, it likely only assists with speeding up deployment. Logon scripts or group policies are the easiest to implement and most enterprise applications today support some level of automated custom installation.

But isn't there more to lowering the cost of managing applications?

As you'll see, there's much more to managing applications (and the associated cost) than just pushing out a new version. The first step is to ensure you're defining application management as a lifecycle. If you're not, and are simply in the mode of deploying applications, then your costs are probably much higher than you realize.

THE ENTERPRISE APPLICATION LIFECYCLE

Each application has a life of its' own within your enterprise from "cradle to grave," with costs associated at each step. So to determine what your costs really are, it's important to understand the lifecycle of an application in order to identify where your greatest costs are when aligned to industry and organizational influences.

Take **Figure 1**, as a simplified example of the application lifecycle for an enterprise, to represent the process IT must go through with each and every application.

With so many devices to support, IT must employ an automated method of delivery of the packaged application.

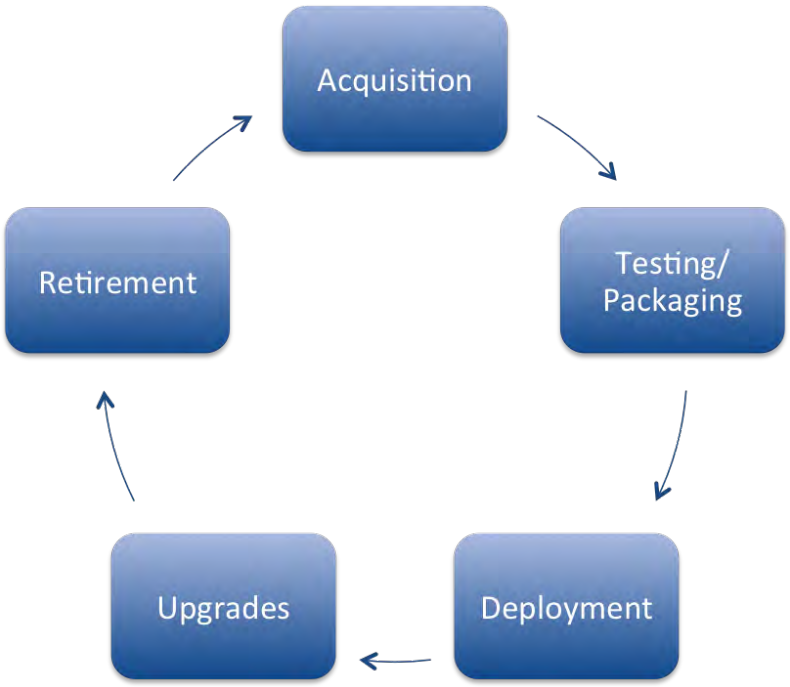

Figure 1: *A Simplified Enterprise Application Lifecycle*

The simplified lifecycle can be thought of in these five steps:

- **Acquisition** – Organizations need to assess the need for – and procure – applications to meet the business needs of the organization. These software assets must be managed throughout the application lifecycle to ensure your organization is always maintaining the right number of applications in the portfolio.

- **Testing & Packaging** – You plan on deploying out to potentially hundreds or thousands of machines, so that requires an ability to ensure a given application doesn't conflict with other applications, versions of applications, and all operating systems it will be deployed to. Once tested, applications are packaged for consistency, a critical aspect of lowering an applications total cost of ownership and improving service quality.

- **Deployment** – With so many devices to support, IT must employ an automated method of delivery of the packaged application. Simple solutions like group policy can be used, but to ensure both delivery and completion of the install, more advanced solutions must be used.

- **Upgrades** – Each application takes on its own life with patches, security updates, and service releases throughout its time within your organization. Like the initial deployment, compatibility, automation, and consistency are key to reduce the cost of support over the life of the application.

- **Retirement** – Applications can be retired from a given device for a number of reasons: reclamation of an unused license, removal of an older application version to make way for a new one, or simply the removal of an application no longer used nor supported. In this phase, automation can be equally as key as that of deployment to keep support costs low.

IT doesn't exist in a vacuum, and there are evolving industry trends and organizational needs that push and pull IT in different directions.

Most organizations have portions of this lifecycle under control. Some even have some automation in place. The goal here is to mature the process from basic packaging done on an ad hoc basis, to being "change ready" where the process is automated and IT proactively plans for major releases as a day-to-day process rather than a huge point in time project. As your application management processes mature, your costs will be reduced and service delivery will improve.

With time and focus, you can get there, and you will lower the cost of managing applications. However, IT doesn't exist in a vacuum, and there are evolving industry trends and organizational needs that push and pull IT in different directions.

So, what's causing the cost of application management to rise?

THE RISING COST OF INCREASINGLY RAPID RELEASES

The availability of cloud-based applications and customer demand for more security, more competitive features, and more value for the money spent, have caused traditional applications and operating systems to have shorter and shorter product cycles. What was once an exercise that only occurred in bursts every few years, the updating of operating systems and applications, is now a part of everyday IT.

Take the release of Windows XP and, subsequently, Windows 7, shown in **Figure 2**, where the red line estimates the amount of IT activity and

An increase in workload exists as organizations focus all their attention to migrate to the next release.

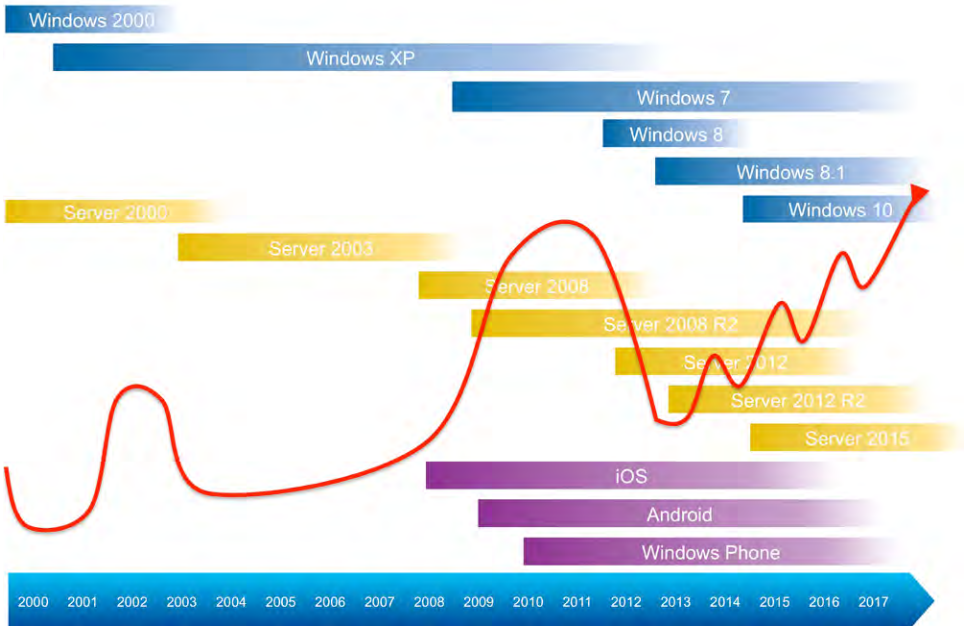

Figure 2: *The Increase in Speed of OS Adoption*

resources required to update systems. An increase in workload exists as organizations focus all their attention to migrate to the next release, but then dies down as they skip releases and wait for the next "big one". With the faster release cycle that Microsoft has committed for Windows 10, along with the fact that enterprises are supporting more operating systems, the IT workload will continue to be impacted.

Internet Explorer is another great example. Microsoft only plans on supporting the most recent version of IE (which currently is version 11) on Windows 7 and 10 as of January 2016. But many organizations are still running and have standardized on earlier versions of IE – usually 8 through 10 – due to compatibility with internal web-based application. Over the years, they simply haven't had the time and resources to update internal applications for every browser version, so it was far more cost-effective to just standardize on one. Given Microsoft's new support stance for IE 11, any organization fitting the aforementioned descriptor will need to make the move to IE 11 by January 2016.

Using patching solutions or configuring automatic updates will address any minor updates, but vendors are putting out new releases far more frequently, also shown in **Figure 2**, raising the question: Are more frequent product cycles actually increasing application costs?

Anyone managing applications has the same simple goal: deploy approved applications to end users when and where they need them in a consistent manner. To accomplish this, new versions and updates of applications must be treated in the same manner as the initial application itself: tested for OS and device compatibility, properly packaged, and published to the deployment system and the enterprise app store. And given the increase in release cycles, this process will need to be done more quickly and more often.

No longer a single vendor approach

Also demonstrated in **Figure 2** is the addition of multiple OSes as part of the norm. Driven by business demand, the enterprise is far more heterogeneous. Mobile devices, and adding in the growth in adoption of MacOS and Android as part of the standard set of client operating systems, has only made IT's job more difficult.

Mobile devices, and adding in the growth in adoption of MacOS and Android as part of the standard set of client operating systems, has only made IT's job more difficult.

And because update schedules aren't synchronized, IT has a new update from a different vendor each week. It can get very convoluted without automation of your testing and deployment processes.

Given the faster releases, and more OSes that IT needs to contend with, this usually results in a lack of testing, where IT just decides it's easier to push out the update or new application and let the helpdesk handle the issues.

Lowering the Cost

To lower the overall cost of the lifecycle, prepared IT organizations are proactively starting to plan for these releases. The lull in **Figure 2** that burst after each OS release (which rings true with major application releases as well) should look far more like lots of activity before the new version is available. They are making plans to standardize on these new OSes and applications rather than just thinking about upgrading to them sometime in the future. Application rationalization projects get underway in advance, evaluating the application portfolio in an effort to reduce the number of internally supported applications and vendors.

The use of automation in every phase of the enterprise application lifecycle is critical to reduce the time it takes to test, deploy, upgrade, and retire applications brought on by frequent releases. As planning begins, automation is a key factor in IT's rapid response to a forthcoming new release.

While organizations can take steps to lower the costs associated with the push from rapid release schedules, IT is also being pulled forward by users taking matters into their own hands.

THE RISING COST OF SHADOW IT

Shadow IT is a problem many organizations have unfortunately brought upon themselves. Users want to innovate faster, bypassing IT and taking the find-it-yourself approach of "there's an app for that", moving the device farther away from a corporate standard. This only further complicates your ability to meet the objectives of ensuring application compatibility, standardization, and ease of support.

Shadow IT is a problem many organizations have unfortunately brought upon themselves.

But, what do they need IT for if they can source their own solutions?

The reality is if you aren't able to automate your application lifecycle processes, create a self-service infrastructure, and essentially modernize your approach to be IT as a service, you simply won't be able to keep up. Oh, and remember you need to do this in a more cost-effective way than you're doing right now.

Lowering the Cost

To become relevant once again, IT needs to change from one that does the work, to one that gives the user a service-oriented experience. It's the difference between waiting for the flight attendant on a plane to finally get to your row and ask you what you want to drink, and using a shopping cart-styled touch screen to add your items, swipe your credit card, and have someone stop by and hand you what you ordered.

See the difference? You can lower the cost of managing applications while engaging users by maturing your process, automating and improving service delivery – all cutting down the time it takes to deliver the right applications to the business. This is what makes cost-effective IT as a Service possible; embrace the user experience and methods users want to interact with IT and assist them in getting to the same goal.

The introduction of an internal, or enterprise app store provides the necessary self-service and instant gratification that Shadow IT provided itself previously, while simultaneously limiting the scope to pre-tested, authorized, and packaged applications. Users feel they have ownership of

the process, when in fact IT is clearly in control providing standardized and consistent applications that meet the business needs, keeping costs low. In addition, licensing and compliance is maintained by the enterprise app store, ensuring licenses are available before an application is deployed, as well as reclaiming a licenses should it go unused.

KEEPING APPLICATION COSTS LOW

While this paper has touched on only a few areas of the application lifecycle, the concepts of automation and self-service woven throughout apply to every part of the lifecycle.

By utilizing automation, every part of the enterprise application lifecycle is managed more cost-effectively.

The increase in applications, vendors, and release frequencies has been noted and taken advantage of by the consumer market, making the need to automate the enterprise that much more urgent. You simply can't match the pace on your own, nor do so and reduce costs.

By utilizing automation, every part of the enterprise application lifecycle is managed more cost-effectively. The speed at which applications can be tested against anything that exists in the organization dwarfs manual efforts, speeding up delivery. Delivery of applications can be self-serviced by the user in less time than it takes to generate a helpdesk ticket, offloading the work from IT and increasing the productivity of the user. Usage can be monitored, assisting with licensing and license reclamation lowering licensing costs.

At every step, an opportunity exists in the enterprise application lifecycle to improve service, speed up delivery, and lower cost. It's only through automation of the entire lifecycle that you will truly mature your processes, begin to be proactive, and have the opportunity to get ahead of the frequent releases and resolute users that are setting the pace and keeping your application costs high. ■

With nearly 20 years of enterprise IT experience, Nick Cavalancia is an accomplished consultant, speaker, trainer, writer, and columnist and has achieved certifications including MCSE, MCT, MCNE and MCNI. He has authored, co-authored and contributed to over a dozen books on Windows, Active Directory, Exchange and other Microsoft technologies. He has spoken at conferences such as the Microsoft Exchange Conference, TechEd, Exchange Connections, and on countless webinars and at tradeshows around the world.

FLEXERA® SOFTWARE