



WINDOWS 10 SERVICING AND THE NEW REALITY

Succeeding in the Fast-paced Now of Windows Updates

By **Greg Shields**



Based on a Fireside Chat webcast featuring Steve Campbell, 1E technical evangelist, and moderated by Greg Shields, Microsoft MVP and industry notable.

I had an opportunity recently to sit down with Steve Campbell, technical evangelist at 1E, to talk about the new reality in the Windows 10 release cadence.

I think all of us in the industry were shocked at how, with Windows 10, a new OS every few years has almost-overnight evolved into a new OS every few months. Our initial shock may have worn off, but many in IT are still trying to get their arms around what exactly to do. I cornered Steve at 1E in this Fireside Chat webcast to ask some very pointed questions about what to expect as we all begin this next phase in servicing the Windows desktop.



Greg Shields: Let's talk a bit about this new mindset, this new reality, as it relates to the new, or really "current" cadence with



Windows 10 releases. I think at first we all were a little shocked at how fast these releases are now coming. That shock has abated a bit, but I think a lot of us here are still trying to figure out exactly what to do.

As I'm aware, there are a couple of different approaches that people can use. On one side, you have the Long Term Servicing Channel (LTSC), which like in previous generations, updates the OS only every few years. On the other, you have the Semi-annual Channel, which as the name suggests, updates twice a year.

In your experience, are you seeing people staying on the Long Term Servicing Channel, or are most folks coming around to the Semi-Annual Channel's increased cadence?

Steve Campbell:



That's a question that pops up a lot, because folks are initially attracted to the LTSC, but that channel was built for some specific purposes. It was built for specific machines that absolutely can never change, and which don't really need any of the new, extra features. For these, you should think like ATMs and single-purpose kiosk machines that don't change a lot over the long term.

I have seen maybe only a single-digit percentage of customers go down the

path of LTSC for the bulk of their Windows 10 estate. A lot of them, specifically in manufacturing, look to put robots and shop floor systems on a long-term release cadence. For the most part, folks are fighting between what used to be called Current Branch versus Current Branch for Business. These have since been renamed to the Semi-annual Channel and Semi-annual Channel (Targeted).

Now that we're over two years in, most folks I work with are starting to wrap their heads around what that Semi-annual Channel means. They realize that OS deployment is no longer a point in time exercise. It's really a service that IT must provide and have available, because it's not just a one-and-done activity anymore.

have. They have to be better at managing their application portfolio and library. For this, they're finding that automation is really their friend.

The old days of hire an army and bring in tens or hundreds of people, stack all the boxes up in a corner, build them, and carry them back to people's desks, those days are gone for most IT organizations. That device, the Windows 10 machine, has become a lot more like your phone in that anybody can perform wholesale OS updates.

I mean, if you can do it for your phone, why can't you do it for your corporate Windows 10 device?

Greg Shields: It seems to me that when you're getting into this new mind-

THE WINDOWS 10 MACHINE, HAS BECOME A LOT MORE LIKE YOUR PHONE ... ANYBODY CAN PERFORM WHOLESALE OS UPDATES. —STEVE CAMPBELL

Every organization has their 20-30 percent of machines that are turning over due to attrition. These are through hardware swaps, regular hardware lifecycle refreshes, and the occasional break/fix. With these, you're always doing some sort of new build. But most of the customers I talk with are resigned to the fact that it is good to stay current. If I am current it is less likely that I'm to get hacked by the Petya, WannaCry, and other malware of the day.

More than anything, IT organizations are finding that they need to get a much better handle on what applications they

set, there's a "first jump" that you need to go through. If you're on Windows 7 or earlier, there's this initial large amount of work to get onto Windows 10. However, then some of that effort can be parlayed into automating the updates beyond that. Can you talk a bit about that first jump, and the lessons learned from that first jump which can be later leveraged for after you're in the more-regular cadence with Windows 10?

Steve Campbell: I'll start by pushing back a bit on the different versions and the real amount of work involved with that jump of which you speak.

Remember that back during the move to Windows Vista and Windows 7, Microsoft made some major, substantial changes to the operating system that broke a lot of applications. You had to deal with the x86 to x64 shift. You had to deal with a new, locked-down OS, the new User Access Control prompts, and that very painful process of discovering all the new additions. These and more tended to break a lot of apps, but they were big, major changes that needed to happen. For the most part, they've now since happened.

The take away from this is that when folks made that jump to Windows Vista or 7, they at that time had to figure out how to take an inventory of applications and they had to get a handle on what user types they had. A lot of that original research and effort is still relevant today, if that organization kept it.

Not having kept that collective information is one of the real problems. Back in a release cycle where OSs might last three or five—or eight or ten—years between upgrades, in a lot of cases that tribal knowledge along with the instrumentation and automation got lost or wasn't maintained in the intervening years. For many customers, they may need to locate and then dust off that knowledge to start the process with this Windows 10 upgrade.

Fortunately, there are a lot of tools in the space, some of which have been instrumented in Configuration Manager, the Microsoft Deployment Toolkit, and other tools, to make this possible. In the third-party space there are still lots of

tools that assist with application readiness, application rationalization, and so on, that still exist today to assist with the problem.

I'll admit that most organizations have a pretty good handle on the number of devices they have. What's usually not well understood is their visibility and control over their applications. Very few of the companies I talk with have an edict or mandate that nothing gets deployed without it going through SCCM or some other third-party application deployment solution. Some orgs have a bit more flexibility in that most apps are delivered through these tools, but small counts of individual apps are still installed by hand. It is that long tail of applications, and their lack of standardization, that really causes problems for people when they get to this point.

If you're like most of the customers I work with, when you get ready to go to Windows 10, you may still have a very Windows 7 mindset of, "I need to identify and test every desktop. I need to locate-and-then-test every piece of software."

That is absolutely not the direction Redmond is pointing people towards. It's not the right story. There will always be some customers who need this approach due to regulatory, compliance, or other reasons. But, if you must test thirty-five different flavors of Adobe Reader, you're not going to get that done in a timely fashion.

Most of the discussions I'm having with customers is, "How do I get those application inventories? What do I need to test? How do I test it? How much automation can I bring?"

These folks really need to start embracing the “ring theory,” if you will, that Microsoft is doing right now with their Insider Builds. Some folks see the updates very early on, and they’re the proverbial canaries in the coal mine. They’re empowered to tell you if their collections of apps feel like they work well. If they do a good job and you’re listening, then you don’t necessarily have to test all three- or four-thousand apps in your entire catalog.

10? If I have, say, a thousand applications, how compatible are they going to be generally?

Steve Campbell: First, a full disclosure: What I’m about to say is very anecdotal, since everyone’s application portfolio is different. Somebody that’s using a higher number of off-the-shelf apps will probably have a higher positive compatibility rate than someone who writes a lot of apps in-house and maybe hasn’t updated their coding practices.

OUR OLD MONOLITHIC AND NON-COOPERATIVE APPROACH JUST WON'T WORK ANYMORE.

—STEVE CAMPBELL

Our old monolithic and non-cooperative approach just won’t work anymore. I mean, look at it this way, if you take one hour out of a person’s day to test each application, that’s a lot of hours, a lot of lost time. Being able to standardize and reduce those app counts, and then automate their replacements when you do a Windows 10 deployment, is going to offer huge dividends.

Greg Shields: You mentioned earlier, and I think we’re all aware, of the efforts Microsoft put into evolving the core OS for Windows Vista and Windows 7. I think most of us are also aware that the subsequent updates from Windows 7 to Windows 8, 8.1, and 10 didn’t involve such massive, breaking changes for apps. That said, though, talk to me about exactly how compatible are apps written for Windows 7 or 8 on Windows

The interesting bit is that as the OS has matured, there’s always been an application compatibility engine built into Windows. That engine has gotten better and better over time. As a result, a lot of the tricks you might have had to do in the jump to Windows 7—like application shims and OS version lying—are in many cases already handled. So, a lot of those old, bad coding practices done by off-the-shelf and in-house coding teams have already been addressed. That has tended to force the compatibility rates up fairly high.

As I look back over the last six or seven years, I can think of only a handful of apps that required a bunch of what I would call “heroic work” to get compatible. In most cases, these were old, crusty apps, often times written on ancient technologies. Virtually all of these were good

candidates for having that discussion on whether the app in fact should be retired.

We know that OSs are going to change, we know hardware is going to change, and we know that apps are written based on the hardware capability that's available at the time. This introduces an important question: That app you brought into the organization 10 or 15 years ago, was the intent for it to run forever, or was there a lifecycle for it? If there wasn't, what can we do now?

For these apps, I typically invoke one of what I call "The Rs": You can remediate it, you can re-platform it, you can retire it, or you can rewrite it.

That discussion is now relegated to maybe 5-10 percent of your apps, instead of being 20-40 percent of your portfolio as it was in the Windows Vista and 7 days. By having that solid handle on how important these apps are, you can focus then on those that get the donuts made and get the taxes done. You can focus on the truly mission-critical apps. Ultimately, you don't want to be applying cycles testing every single version of every single app that's out there.

I'll add too that it's important to realize that one's "problem apps" a lot of times aren't actually apps that won't work. Rather, they're problematic because they won't install. For these, it's not so much that the app won't work, but the app's installer won't get it installed. In some cases, that installer didn't know what to do when it saw a Windows 10 OS footprint. Others would ask for Administrator rights and then didn't even need them to get installed. A lot of apps that get flagged as

problems can, in fact, be easily fixed through simple tricks like version lies and other shims which are a function of Windows' application compatibility toolkit.

Greg Shields: This is a conversation ostensibly on the OS upgrade to Windows 10, but it's funny how our talk on the OS upgrade spends a lot of time focusing on the applications. Earlier you talked about the idea of "automation" as it relates to the testing of applications, can you dig a bit further into what you mean by testing automation and how we can get to this point where we can have the necessary apps get tested.

Steve Campbell: We've seen two major approaches, particularly when you've got a lot of applications with single- or low-installation counts. These are the one-offs for which you may not even know how they're used or what they do.

On one side, you can automate the testing. You do this by standing up virtual machines with Windows 10 and invest in a configuration management solution to create test harnesses for every relevant combination of applications related to a role. You then assign a set of users to test those combinations within that environment.

This involves identifying the user roles who need the apps, creating the test beds, and then using automation to provision VMs and deploy the apps to test each new OS version. The problem with these is that, at their core, they're only slightly better than the all-encompassing "we're going to test everything, everywhere" approach that we've already discussed doesn't scale all that well.

The alternative approach, and this is the approach I'm seeing more customers move to every day, is to follow that ring process we talked about earlier. With it, you identify a few users to be that insider ring, that Ring Zero, inside the company. Give them early access to each new OS build, and then set them loose to just do their everyday activities.

whole world get it. The intent is that somewhere along that three- or four-gate release cycle, somebody should be raising a hand to identify application problems before they hit the next, larger group.

Smart companies are essentially emulating these rings with their own applications to limit the impact of

ONE QUESTION I THINK IS ON A LOT OF FOLKS' MINDS HAS TO DO WITH WHICH BUILD UPGRADES TO ACTUALLY APPLY. —GREG SHIELDS

This may or may not be on production equipment. They may have a separate set of equipment, or a separate VM, to perform their tests. Since they're automatically going to get each new OS build, they're pretty quickly going to see if that update breaks things or if it doesn't.

If this small insider group sees it early, they're usually only 3-5 percent of a user base, they can let IT know before the update gets rolled out to the next, larger group. IT can always roll them back if the update significantly breaks the experience. However, they're your "canary in the coal mine" because they're empowered with being the first ones to raise their hands and say, "I've got a problem."

This approach is basically what Microsoft is doing with their own Insider releases. They have their own small group of folks who real-world test each release before anyone else sees it. Then, after the first group approves it, the release goes slightly broader, and then later does the

incompatible applications before they hit larger groups of people.

Greg Shields: One question I think is on a lot of folks' minds has to do with which build upgrades to actually apply. The messaging from Microsoft seems to be that one should plan for and execute every build upgrade. Are you seeing organizations apply every OS upgrade, or are they skipping some versions to put some space between these major revisions?

Steve Campbell: Some organizations are absolutely skipping some build upgrades. Some are watching to see what their industry leaders are doing. To some extent, non- and quasi-technical externalities like the size of the organization and their own politics and security stance come into play.

I know one large insurance customer whose certification process to release products to their users requires 18 months. They need to test every piece of software with every known configuration

and be able to say that every group of apps is certified. That process, the way they're doing things right now, takes them 18 months. I can say that they're not going to be able to keep up with the twice-yearly cycle. This company, by virtue of their decisions, are going to have to skip build upgrades.

Some folks, because of their in-place agreements, just aren't going to be able to update every six months. I know of another organization that, due to the criticality of their systems, has to have a third-party verification, an external vetting process to certify their systems. That obviously also takes longer, and so this group is going to be in a slightly different cadence.

SOME FOLKS, BECAUSE OF THEIR IN-PLACE AGREEMENTS, JUST AREN'T GOING TO BE ABLE TO UPDATE EVERY SIX MONTHS. —STEVE CAMPBELL

I think that smaller customers are probably going to be able to go more quickly. Some organizations upgrade one group at a faster cadence, while retaining others at a slower pace. These orgs typically identify a technology adoption group to be those advance troops. Often times these individuals are sales folks or others that are either externally facing, or who spend a lot of time outside the brick-and-mortar in less-secure and more-mobile situations. These folks get put on that more-current cadence due to their increased risk profile.

Greg Shields: A final question, which I hope is a question everyone here is

looking for. Let's say that I'm on Windows 7 or one of the even-earlier versions.

What is the right timing for making the jump to Windows 10? What is the right timing for starting that project to get onto that new cadence?

Steve Campbell: Well, frankly, how soon is now? Windows 10 is two-and-a-half years old. If you haven't even started the planning yet, then why not? If it's not enough people, not enough cycles, or not enough money, you can at least start the thinking and the planning now.

A lot of what we've learned in previous deployments still applies. You still have hardware to think about. You've still got application inventory to gather and applications to test. You've got training to do

for help desk and IT. You've got vendors to work with. Frankly, if you're not on [the most current version of Windows], most of your vendors likely already are.

All of these other behind-the-scenes activities can still be done. At a minimum, start getting to know your users and where they're at. This much is always true a couple of years after a major OS release: Your users, if they're buying PCs for their homes, guess what OS is already installed on them? It's very likely Windows 10.

Your users are already on an OS that's more advanced than what they have at the office. They're already used to it, so

AT THE END OF THE DAY, IF YOU'RE NOT ON THAT
NEW OS, DO YOU WANT TO BE THE PERSON WHO
HAS TO REPORT YOU GOT HIJACKED BY THE
NEWEST VERSION OF WANNACRY. —STEVE CAMPBELL

that training gap is likely already gone. They've already seen and gotten used to File Explorer, Edge, the Start Menu, and all those other look-and-feel things that come with a new OS version.

Start now. Find out who those power users are. Use your inventory tools. How many of your departments have a "shadow IT" of people who've already secretly upgraded to Windows 10. Recruit them. And, then, if they're already running Windows 10 and they're already running your applications suite, then guess what: Those applications are already certified. This approach actually works.

The more you can get people to dip their toes in the water with you now, the less intensive, costly, aggressive testing you'll have to do down the road. There's probably some infrastructure in need of upgrades. You've probably got some devices that aren't going to support it, or aren't going to support the security features, but you can start checking that inventory now and figure out early what needs a hardware upgrade and what's ready for that new Windows 10 OS today.

At the end of the day, if you're not on that new OS, do you want to be the

person who has to report you got hijacked by the newest version of WannaCry—or whatever malware—that wouldn't have been an issue if you'd upgraded earlier.

Ultimately, what's the cost of doing nothing?

ABOUT 1E WINDOWS SERVICING SUITE (WSS)

The 1E Windows Servicing Suite (WSS) (<https://www.1e.com/products/windows-servicing-suite/>) is designed to directly address the challenges of migrating to Windows 10 from a previous Windows version. Unlike some competing solutions, 1E has not designed WSS to replace your existing management tools. Instead, WSS augments Microsoft's System Center Configuration Manager, so as to extend its capabilities.

Find out more: <https://www.1e.com>

