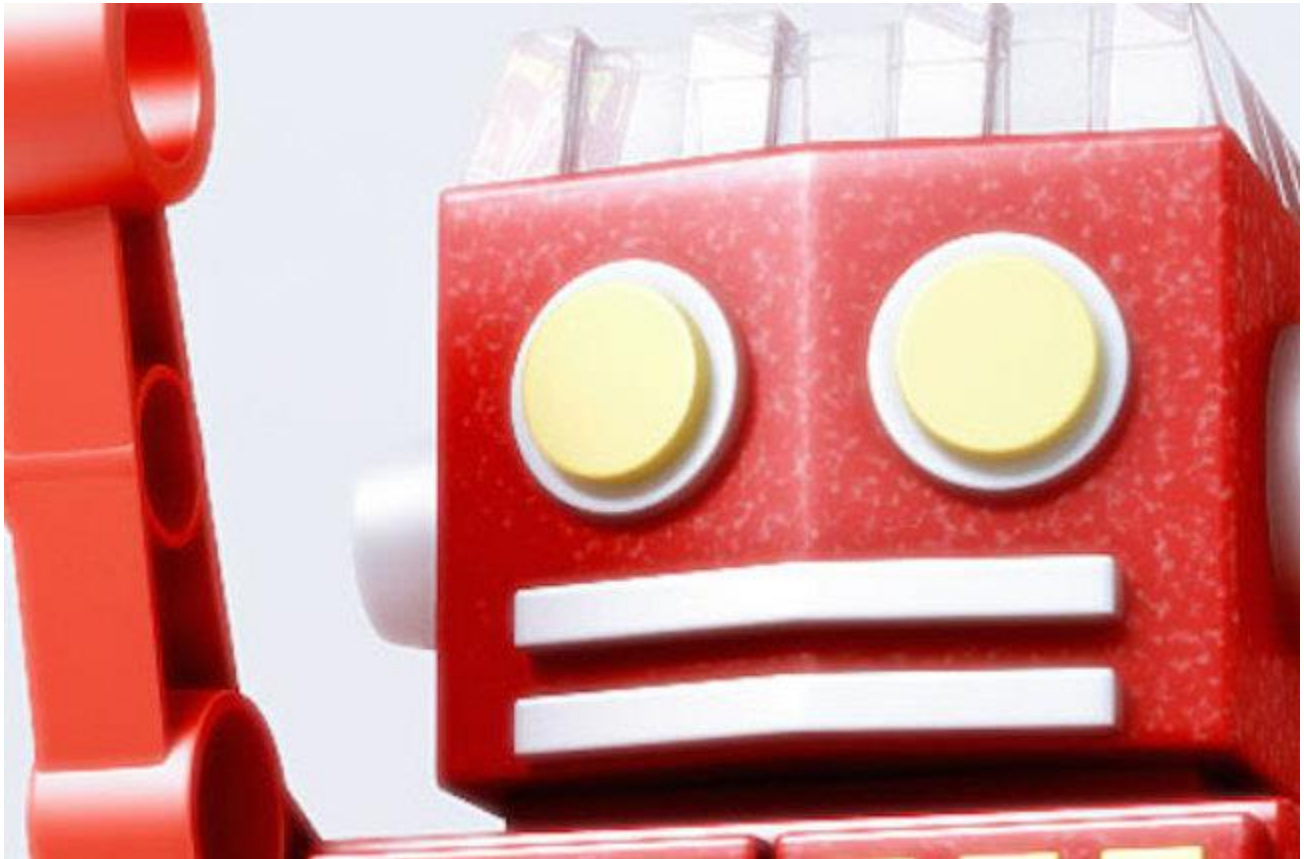


# The future of IT is – to deliver automation. Discuss

Adapt or die, says Trevor Pott



1 Sep 2015 at 10:11, [Trevor Pott](#)

**Sysadmin blog** You don't have to be a large enterprise to benefit from technology, though access to seemingly endless resources tends to help. I've worked in SMB IT my whole life and automation changes *everything* at this level.

So many things that reasonably should be automated simply aren't in the SMB world. We rely on fallible humans. Forgoing any discussions of speed gains - automation may or may not actually produce them - the real benefit of embracing automation is the reduction in mistakes.

Reading a postal code from a piece of paper and entering it into a computer is easy. Most people won't make a mistake if they have to do it once. Do it 1,000 times a day and the odds of a mistake go up enormously.

Entire fields of management, psychology and so forth emerged to understand how and why humans make mistakes at repetitive tasks and how we can prevent them. True nerds look at the problem described above and solve it with computers.

## **Automation 101**

This is easy if the information that needs to be put in the destination computer is arriving from another computer in a predictable format. Write a parser to convert the data into something that the destination computer expects to see and inject the data in an automated fashion.

If the destination computer refuses to allow data imports you can reduce human mistakes quite a bit by simply adding bar codes. Instead of reading the information and typing it in, the information is a series of bar codes to be scanned. Bar code readers can pretend to be keyboards so it's easy to sneak one in, even if the computer in question is supplied by someone else (such as the shipping company).

The more all-encompassing the automation, the bigger the returns

This is automation 101. A few hours to a few weeks of work on its behalf and tens of thousands to potentially millions of dollars a year in shipping errors simply go away. The simplest such system I built was 33 lines of PHP that was executed every 15 minutes as a cron job that took me about eight hours.

That cron job parsed an incoming HTML file and both injected the information into an internal database and printed a sheet with barcodes. It ended up saving the company about \$60k in the first year.

The more all-encompassing the automation, the bigger the returns. A video rendering system I worked on had big problems with scalability because everything was done manually. Each node in the render farm had its own operating system that was installed by the company's systems administrator.

All operating systems and applications were patched one at a time and hardware failures meant lots of downtime for a node, including possibly a complete rebuild of the operating system and application install.

Simply introducing the concept of [system imaging](#) saved enough time that the company was able to double the size of their data centre without needing to add a sysadmin.

It was during the next refresh cycle that we really saw savings. Instead of loading operating systems onto each server we pushed out operating systems to the nodes on boot.

Each time a node started up it would ask the central server for an OS, grab an image and be ready to go in less than five minutes. Updating systems was as simple as patching the central image and rebooting nodes.

Up front, there was a lot of extra effort put into the automated central system, but the end result was a data centre that practically ran itself. The sysadmin didn't have a lot to do except update patches and swap out nodes when he experienced hardware failure.

On the next refresh, we even automated out the patching of the centralised images.

## **Automate or die**

The discussion about automating systems deployment versus babying servers individually has been done so many times it's now known simply as the [pets versus cattle](#)\* argument.

Boiling down years of debate gets you the simple axiom: automate or die. No company, no matter how small, can survive if its competition can offer the same goods or services at lower prices.

Pets versus cattle evolved mostly out of a discussion about hyperscale technologies. The sorts of things you'd see in a public cloud, or in a private cloud like HP's [Helion](#). It has traditionally been an enterprise and service-provider level argument, but in truth it applies to businesses of all sizes.

In the western world, we aren't going to get those lower prices by driving down the cost of labour. Globalisation means everyone more or less pays the same prices for raw materials.

If we are to survive in an increasingly globalised, hyper-competitive world we must reduce wastage of materials and the need for labour as much as possible through automation.

It isn't enough to simply ensure that operations are working today. To survive, businesses need to continually look at every aspect of operations and ask "is this going to be good enough in five years? Ten? Fifteen?"

The future of IT isn't beating printers into submission or explaining to Sally from sales where to find what's she looking for on the Ribbon Bar for the eleventeenth time, it's delivering automation.

Those who fail to adapt - from IT practitioners to vendors - are as doomed as the companies that pay them. ®

---

Talk to Triangle Computer Services.

We can probably help you.

[ian.kc@triangle.ie](mailto:ian.kc@triangle.ie)

Tel: +353 1 657 9700.