INFORMATION TECHNOLOGY

CAUTIONARY TALES

SCOTT CARMICHAEL

Information Technology Cautionary Tales

Published by Scott Carmichael

Cover Art by Scott Carmichael

ISBN: 978-0-615-58087-6
(Part of Scott Carmichael's IT Manual: Topics in Technology, Linux Web Server Support, Information Technology Cautionary Tales and Help Desk Support Fundamentals)

Information Technology Cautionary Tales: An Introduction

Over the years, I've witnessed first-hand and heard through others' stories about all sorts of IT-related disasters, from people knocking over expensive servers to employees getting their brand-new laptops stolen right after getting them to non-IT departments making IT decisions and screwing everything up.

Obviously I can't (and won't) document everything that I've learned about, simply because it would probably make me sound like I have nothing good to say about IT. That just isn't true. But the reality is, bad things relating to IT happen frequently at companies, and IT professionals should always be on the lookout for red flags that may alert them when something is very wrong, or even if it's time to move on.

These selected stories highlight some of those cautionary tales I've come across and heard about in the past.

I hope you enjoy the book.

About the Author:

Scott Carmichael is an award-winning, professional Web Developer, Designer and IT Professional with degrees and certifications in both Visual Design & Web Media and Information Technology. *You can learn more about him and his work by visiting http://www.scottcc.info.*

Table of Contents:

Chapter 1: Leaving IT In The Past

The software company had been around since the 1990s. It wasn't in Silicon Valley, but rather in a state and area of the country not known for much of anything related to "com-poo-ters," as the locals might refer to them. No, this was a different place – a place some have said exists about ten years culturally, economically, technologically and socially behind the rest of the nation.

But what this particular software company had done was find itself a niche that had some demand, with founders and investors who figured that when government regulations would almost certainly take effect (at some point in their particular field), business would boom.

And about a decade after it was formed, it finally found success. It had so much, in fact, that the company ballooned to over 400 employees and took up three parts of an office complex and another location across the city. Around 2010, the business was thriving, and the owners wanted out while the company was at its peak. They were smart.

Over the next couple of years they courted buyers and eventually found themselves acquired by a larger competitor a state away. And within a couple years of that, that larger company was bought out by an investment company who already owned an even bigger competitor in the region. Suddenly, in less than half a decade, this small 400-employee business was part of a nearly 2,000 employee-strong software industry behemoth.

But that's where the fairy-tale ends.

It turns out that the owners of the 400-person company never really thought ahead, and the companies that took over IT affairs for the acquired business didn't either. Nearly half a decade in, the original niche business was its own island technologically – a red-headed stepchild that its larger siblings simply didn't bother to worry about.

Years earlier, in the early 2000s, when Sun was still Sun and Oracle was not even in the equation, the IT staff (at the smallest, niche company) had decided that instead of buying its 400-employees actual computers, they would only buy some of them real PCs. The rest? They'd get a "Sunray," which was effectively a cheap dumb terminal that only had one decent feature: A large flatscreen display.

The problem was, well over a hundred employees had these Sunrays, and instead of logging into their very own computer each day to do their intricate billing-related work, they'd have to spend their time remotely connecting to a Windows Server session. Even worse, their profiles were stored on two entirely different Windows Server instances, and a bizarre profile transfer system had been poorly rigged up between them.

Log-in to your Sunray in the morning, and a file you saved in your Documents folder would be there. Log-off for lunch and back in that afternoon, and you'd be on the other Windows Server (because only so many users could be on any given server at a time) and there's a good chance your file wouldn't be there.

So, employees were told to effectively ignore “local” storage all together. Instead, use their “shared” drive, just in case their Windows profile had issues. And boy, did it ever have issues. The most common of which came from Outlook, which freaked out over temporary storage paths and roaming profiles … the solution to most Outlook issues was to simply blow away their profile and re-download all of their mail. Sure, employees had access to webmail in their browser in the case of an emergency, but these people viewed PCs as a tool, and were largely terrified of any change to their daily routine.

Printers were even more bizarre, because since there were two Windows Servers keeping everyone up and running, the printers had to first be recognized and installed on those. You'd think that then would cause printers to be visible and available to all logged-in users … you'd be wrong. Fairly regularly, networked printers would simply get confused over too many queued jobs and would suddenly stop responding for random chunks of users. Logging-in and off didn't help. Switching servers didn't. Eventually, IT techs would have to manually restart the Windows Server, kicking everyone offline for at least 20-30 minutes while Windows got its act together and the printer drivers could be uninstalled and reinstalled.

Employees also often complained about the lag (because they were remotely logging into a machine, anything animated like a video was jerky and unwatchable) and the sound output on the Sunrays didn't work either. As multiple widescreen monitors started to be handed out to the normal PC users to make them more efficient and help them with their workflow, the Sunray users became jealous. Why were they jealous? Because the Sunrays simply didn't support a secondary monitor for an extended display.

By the time the 400-person company had been acquired twice, it was down to less than 200-people and about 75 still used Sunray devices. Why were these relics from nearly a decade earlier still in use? Because the original owners, the first acquiring company and the large investment company all hated the notion of getting rid of a hundred dumb terminals in favor of standard company laptops that matched everyone else's machines.

It didn't matter that thousands of dollars per month were spent on unnecessary support for a Sunray-based environment when the company that made the devices (Sun) had basically gone under six or seven years earlier. It didn't even matter that to even make the Sunrays communicate with the Windows Servers, all traffic had to be routed through a Sun/Oracle server designed specially to accommodate such a design. So factor that particular server technician's annual salary into the mix as well.

No, their solution to stay with what worked was far simpler (in their mind) than just buying everyone a $500-600 laptop and maybe a $150 extra display that would last them for years (and even allow them to work remotely, if necessary).

Chapter 2:
A Learning XPerience

The geniuses who made the company's inventory control and order system were just that – geniuses.

They knew exactly what they were doing in the mid-2000s, designing a system that would keep their client firm in their clutches for years to come. This third-party software house used the oldest trick in the software-world book – planned obsolescence – to get the business where they wanted: Hook, line and sinker.

Then the third-party company vanished off the radar. What happened to it? Did it file bankruptcy? Did its founders run off with the money? Did a bear devour everyone in their offices? No one knew. The only thing people at the retail store's corporate office knew was this: That software was the backbone of the entire company.

That software had to survive at all costs.

That's why, in 2013, when most of the world was already getting off the old- and barely-supported Windows XP operating system, this business was still very much using it. It had absolutely no desire to change the operating system that most of the corporate office relied on. Even worse, the company had zero desire to even upgrade employees to

Windows XP Service Pack 3, which had been released years earlier.

Was this outright defiance? A spit in the face of the tech world? A brazen and bold way of telling Bill Gates and Steve Ballmer enough was enough?

No. It was all due to something called "J Initiator."

Remember those "geniuses" who designed this mission-critical system for the company? Well, apparently they were big fans of a few volatile things – Microsoft Internet Explorer's Active X Support, Java, Java Applets and Windows XP win32 Support.

The problem was, all of those things radically changed as software needs and operating systems evolved during the 2000s. Windows Vista and onward suddenly stopped letting software do whatever they wanted with memory access and system permissions. You had to spend more time developing software that respected system resources and security settings.

Regarding Active X, everyone pretty much gave up on it when Flash became the defacto multimedia standard and Active X exploits were often named as the cause of virus/malware attacks.

Java – the once fabled unifying platform of everything in the late 90s – had been demoted to "Awww, that thing!?" status by the mid-2000s. Java Applets, which ran in a web browser, were even more hated, because they often didn't like any browser except Internet Explorer.

And so, the company was stuck with using a particular Java version – 1.4.2 – in a particular web browser – Internet Explorer 6 – on a particular OS – Windows XP SP 2 (because SP 3 removed IE6) … all of which were needed to run something called "J Initiator," which would connect to the company's inventory system and let employees deal with orders.

Antivirus software and other standard security software also caused problems, so none of the machines had anything on them, aside from Windows Defender, where it was shown to not cause any issues. The vast majority of workers though had no protection at all from anything. They also couldn't turn on Windows Updates at all, because those automated updates would cause configuration issues.

It was pointless to even get employees new equipment, because XP-downgrade licenses wouldn't help … because on top of all the bizarre system requirements listed above, an even uglier truth was real: The company that configured all of these systems to run properly and connect to the inventory system … was no longer available either.

In other words, if Bob in Purchasing had an operating system freak-out that required his OS to be restored from scratch, they could never load his inventory system on his machine again, even if they had all the software requirements met. The most they could do was find him another machine with the software already on it, and let him use that.

It was a wonder the entire inventory system design didn't bring the company to its knees, long before it ultimately went under for other reasons. A single virus attack or disaster in the office (like a fire) would have crippled the entire operation to the point of people needing to handle orders by phone and using catalogs and pen-and-paper just to get anything done.

Oh, and one last thing – a critical inventory system was actually created in an Access 97 database, way back in the 90s. Only one employee had access to it, and only they had Access 97 on their machine. The company never moved the database contents anywhere else even though it was in use for well over a decade. And before you ask, “Was there even a copy of Access 97 that the company owned that could be used to reinstall it?” No. There wasn't. The employee and the people before them simply copied the program's files from one machine to another, and hoped for the best each time.

Chapter 3: Out Of Time(Clock)

The company's IT department was known to be long-in-the-tooth and barely supported financially. But years of neglect finally caught the management (who approved department budgets) by surprise, and this time they couldn't weasel their way out of the inevitable. They needed a new timeclock system for over fifteen hundred employees, and they needed it yesterday.

Unsurprisingly, their gut reaction was to kick the can down the road. And this had been done many, many times already. Unfortunately for the folks in charge at this point, the can was encased in concrete, and the managers were wearing flip-flops. No amount of kicking would give them a way out of the decision.

That didn't stop them from trying though, and for the first four-months of new owners and management taking over, the timeclock project wasn't even mentioned to anyone in IT. Two months before the end of the year, the company's lone programmer was finally told what he had to do: Make an entire timeclock system that not only mimicked the look-and-feel, but also the abilities of the current RedPrairie timeclock software the giant retailer was using. It also had to be completely compatible with payroll and HR information. This meant that the timeclock wasn't just logging punches, it was calculating everything relating to hourly workers, behind-the-scenes info, and was the key component in the entire payroll system. If it didn't work, or worked incorrectly, well over fifteen hundred people could be without pay.

Oh, and it had to be completely finished and rolled out to several dozen stores in four-months (February).

The programmer – who had been hired only because he was friends with the IT manager who (was on his way out by then) – had no intention of meeting that goal though. No, he was going to sit on his $80,000 non-tech-hub-area salary that was tens of thousands above the regional wage … and do nothing. Why? Because he figured with his manager friend being forced out due to incompetence and poor-planning of his own, the programmer friend was likely next.

He didn't tell anyone that obviously, and for the next two-months, he did the absolute bare minimum the project required. At the end of December though, when the HR department and owners were presented with what the programmer made, they were shocked. Not only was it far from being completed, it also looked completely different and had almost nothing in common with the RedPrairie timeclock all employees and managers were accustomed to using.

In place of that the programmer had decided to build a standalone .NET-based timeclock application that mainly just tracked punches at a local level, and then would communicate with the HR department at the corporate office once in a while, throughout the day. The interface looked like some Web 2.0 glossy/reflective button-filled abomination that any Indian outsourcing team would come up with, by cobbling together random bits of Visual Studio code. It was downright awful, and even non-programmer HR and company owner people knew they had been fooled. From then on, the programmer had all of his remaining programming jobs taken away, and he became a glorified IT server admin for his remaining time there … and they fired him in the second quarter of the next year. But he was okay with that –- he had milked over $70,000 in pay from them by the time of his firing and had only needed to complete a couple tiny projects early on.

But now the folks at the top had a real crisis – they had just two months now to get a timeclock system in place, or else the RedPrairie multi-year contract would have to be renewed, and they'd be out hundreds of thousands of dollars, with at least a six-figure cost being needed by RedPrairie upfront.

Their only option internally at this point was the guy who handled the website coding, and who had made an employee review website for the owners a month or so earlier. They didn't view him as a real programmer, and they certainly didn't pay him anywhere near programmer wages … but they figured they could possibly avoid a huge expense that would severely hurt finances in the short-term by having him focus on it non-stop for a couple months. So, they explained the project, and the web guy eagerly took it on. In his eyes, this was his chance to get a decent bonus of $5k-10k, and he would be even more important to the company.

While the timeclock design itself was simple enough, the real problems came in the form of four things: 1) It had to have the look-and-feel of the existing system. It didn't have to be exactly the same, but it had to be close enough. 2) It had to accommodate over a dozen union- and store-specific sets of rules regarding overtime, benefits, wages, etc. 3) It had to feature detailed reporting and scheduling features for managers at the store level. 4) It had to not only accept an import of employee data for thousands of workers at the store- and corporate-level at any given time, it also had to export tens of thousands of weekly entries to a format a third-party payroll processing business could automatically process.

The design of the web guy's timeclock system was radically different – it ran in a web browser, and changes could easily be made to the core code at any time, vs. worrying about deploying and updating Windows applications to stores. On the downside, PHP & MySQL were slow when dealing with crunching tens of thousands of complex tasks, and often the weekly payroll export for dozens of stores would take 15-20 minutes (just to generate lengthy text output of employee entries).

Six-weeks later the timeclock system was working, and HR and upper-management were pleased and quite shocked it had worked out in their favor. Behind closed-doors, they were high-fiving each other, because $100,000 immediately saved was $100,000 immediately open for bonuses and “company” purchases for themselves. The timeclock system rolled out to well over a dozen stores in the next few weeks, and there were almost no glitches. Any odd things that did come up were due to bizarre union-rules that over-complicated everything in the code.

The story didn't end on a good note though because while working with HR data, the web developer had no choice but to see and work with employee wage data. When he discovered he was being paid tens of thousands less than the programmer on staff who everyone knew now did no programming, he was furious. He asked for an immediate pay increase in hopes that management would match the programmer's rate, while never actually saying he knew what the programmer made (they would get mad). The management took the request as an ultimatum, and not only dismissed it, but also made it clear the web developer should simply feel lucky they gave him a job in the first place.

That same night the web developer started looking for work elsewhere – and two weeks later he was working somewhere else. The timeclock system, only weeks old, was now completely unsupported, aside from a couple short guides the employee made and his comments within the code. The upper management only cared about avoiding paying $100,000+ by March, and didn't care about workers with talent and a good work ethic at all.

Chapter 4: The Cost Of Doing Business

The large bank had gone through a massive merger a half-decade earlier, and now was so large in the region that it had multiple, giant office complexes in an expensive business park that were devoted solely to banking technology. Since the passage of Sarbanes-Oxley in the early 2000s, banks were under more scrutiny than ever before, and with only a few years after the Great Recession having passed, the bank – which had received a TARP bailout – was trying to save money wherever and however possible.

Sadly, one of the most overworked and under-appreciated departments of all, which serviced the 10,000+ employee business, was its own internal Help Desk department.

You see, the management had decided years ago that a call-center-type design was the best one for the company's Help Desk, and that in place of written-tickets and written-replies through a standard-ticketing system found in most large companies, they opted to handle every IT request by internal staff … over the phone.

As you can imagine, at a company with 10,000 employees across dozens of locations and with employees in several states and multiple timezones, this was an incredibly bad idea.

They also decided that everyone's time except for the Help Desk staff's was important, and therefore enacted a strict 15-minute limit on every single call. No matter the IT problem, it should be resolved completely in under 15-minutes. Of course not every single issue could be resolved in 15-minutes – that would be impossible. So, they made a small caveat – a small percentage of Tier 1 tickets could be sent to Tier 2 for expanded support requests, but if the Tier 1 employee passed on more than 5-10% of their monthly tickets to Tier 2, they would be flagged for "serious review" and face warnings … and quite possibly termination.

The management's next brilliant move was to ensure that all people calling in were verified to be who they said they were … to the nth degree. The bank already had strict rules about things like bringing in your own mouse from home (possible termination offense!), plugging in a USB drive (likely a termination offense!) or using someone else's credentials (almost surely a termination offense!), but the approach to employee verification was ridiculous. This was a lengthy process, which almost always took up five-minutes of the 15-minute window allowed for a call. It also didn't help that the Help Desk machines were so bogged down with monitoring software that even the internal ticketing system, command-prompt actions and web browsers available were slow-as-molasses too.

Because of the down-to-the-minute focus by management on Tier 1 staff, they also had a zero-tolerance policy regarding being late. Despite staggering the dozens of Tier 1 workers' schedules so that more than enough employees would be present during multiple timezones' business-start and business-lunch times, they were always watching the clock. Even being three minutes late was worthy of not-only a write-up, but a manager waiting for you, in your cubicle, arms-crossed. Tier 1 employees also couldn't even take a bathroom break without permission. It was surprising that no one there never wet themselves and continued working. Perhaps they did, but they built in adult diapers into their monthly expenses.

RSA key fobs were also a “brilliant” choice by senior IT staff. To verify remote employees were who they said they were, these little keychain devices (which had an LCD-display on them) would show a new set of numbers of every 30-seconds. While calling in, their value had to match what the Tier 1 worker would see, otherwise no one could help them. Considering the fact that only manager-level staff were allowed to work remotely, you can imagine how often they either lost or forgot their key fobs – Hint: All the time. So, the IT geniuses high up devised another plan – a virtual RSA token generator, which was installed on their work laptop. The same verification process would occur, with one difference: The RSA token generator software had to be properly set before the employee left the company's network. So, for example, if the VP of Mortgages forgot to configure his software to work remotely while they were still in the office Monday, when they were at the resort for a conference Tuesday and needed support, we couldn't verify his identity or connect to their machine.

It was an absolute nightmare for staff, and as you'd expect, of all the employees requesting Help Desk support, management-level workers had the worst attitude ... simply because they could get away with that behavior.

Speaking of behavior, despite all of the monitoring, call-logging and employee badge access points everywhere, one thing was hilarious: The Help Desk management were terrified of IT ticket notes' content. You see, one thing they had no control over was erasing or editing ticket notes made by Tier 1 staff. The upper management didn't trust IT because they assumed IT would retroactively alter tickets to boost their dept. numbers (this must've happened before).

Not only that, but people high up in the bank could see everything in the IT ticket database, and apparently word had come from down-high in the IT dept. that no messages about staff behavior, actions or dialogue should ever be included in notes. For example, when a particular manager called in one time and was very rude to support staff, his words and behavior were detailed in ticket notes because that resulted in complications to the ticket (that would make the support technician look bad during review if not included). Instead of being rewarded for writing accurate notes, the Tier 1 technician was unofficially reprimanded for doing such a thing. The threat from management was clear – do that again, and you won't have a job where you're writing ticket notes at all.

Despite all of this nonsense, the Tier 1 staff didn't have a horrible amount of turnover like you'd expect. Sure, about half the staff quit/were fired annually, but for a call-center job like this, one would expect the total to be closer to 80-90%. The trick the other half had learned though was this: Simply do an average-at-best job on 99% of tickets they touched. Because the department had devolved into a numbers-game, the trick to ensure job security was simple: Keep your numbers up. The way to do this was also simple: Make sure every call was under 15 minutes, no exception. Now, the way to do that was where things got shady: Whether the employee calling in received the proper help or not didn't matter – as long as they received approximately what they they needed, that was good enough. Another loophole technicians found to either deflect hard-to-solve or longer-than-15-minute issues was this: Make them call back. The tech would purposely have them do step 1-of-3 (instead of all 3) and say the employee needed to restart their machine and call back. This would not only keep their numbers up, but would most likely cause another technician to do the remaining, lengthier work that couldn't be deflected when the employee called back. It was a slimy environment.

The one thing the company did do “sorta” right, however, was making sure new hires were exposed to lots of issues and how they were handled, long before they ever started taking calls themselves. This would let them hear how other technicians would troubleshoot problems, while the new hire listened on their headset to the phone conversations taking place.

The only real downside to this though was that this training period lasted six-entire-weeks, vs. just one or two, and by the time new hires had their first, non-shadowed call, the "newness" of the job was gone.

They already felt defeated and knew how the support queue and workflow operated, and had little motivation to be enthusiastic about their day-to-day tasks. This also meant the management had little tolerance for mistakes, because in their mind, the new hires should have become experts over six weeks of observing others. The entire training process had become a mess, and no one even attempted to change it for the better. Instead, the management simply hired new people on a monthly basis to replace the same number of people quitting each month.

The revolving door of Tier 1 staff at the financial institution meant that the bank – which may have been good in financial aspects – was absolutely terrible in the most basic of IT work.

Chapter 5: Doomed To Repeat IT

The large retailer hadn't bothered to invest in its IT for years. In some ways, nearly a whole decade. XP was at the end of its lifecycle, and companies all over the world were racing to finally make the leap to Windows 7, which had already been out for a number of years at that point. But this business wasn't racing.

No, this business was doubling-down on its IT strategy: Nothing.

Walking through the corporate office was like going through a museum or place stuck in time. Everything from the walls (which were literally splitting apart) to the carpet (stained by hundreds of “Oh my coffee! Nooo!” moments) to the furniture (last updated in the early/mid-90s) was about a decade or two behind where it should have been. So, in that aspect, I guess the fact that the owners hadn't bothered to invest in IT either shouldn't have been much of a surprise.

The overwhelming majority of employees had two things in common: A Pentium 4 CPU and a 4:3 ratio LCD monitor. Why these two things seemed to be with everyone – even though Intel had stopped making Pentium 4s about seven years earlier and widescreen displays were the norm for half-a-decade by then – was anyone's guess. I'm sure the ugly truth was that cost of new equipment was the reason, but just look at your computer you use for work: Don't you think about 3-4 years in it would be ready to be replaced? What about 5 years? Certainly. 6 years? C'mon now, just give me a new machine. 7 years? I'm not even asking anymore.

And, it should be noted that this company made millions in profits annually.

When the company needed a machine for a new, non-management employee, they typically didn't reach out to CDW or Dell – no, instead they sent the IT staff to the actual basement, where a lot of old machines had been stored. Most of the machines down there had been gutted – for a motherboard here, or RAM there, or a video card here – almost nothing was ready-to-use, as-is. So, instead of spending even $300 for the cheapest Walmart Electronics Department Acer tower, the company instead figured they would simply make an IT support tech devote multiple days to cobbling together some Pentium 4-era monstrosity. Also, when you factored in their hourly wages, those likely equaled the cost of a new low-end PC.

Once in a while the techs would get lucky and find a Core 2 machine … and once in a great while an old, returned i3 laptop would come across their desk when a salesman would quit – but by and large, new hires got a nearly ten-year old Pentium 4 PC. And because older RAM had to be ordered, and that was frowned upon, RAM was precious and had to be carefully divided up. Most people got 1.5GB of RAM, while a select few got two 1GB sticks. Because the company was on XP and it was actually pretty lightweight, RAM didn't need to be massive, but 1GB was pushing the realm of usefulness, especially if a new hire needed to open a web browser and, say, Outlook and Word simultaneously.

At that company, a legitimate solution to resource issues – due to the RAM shortage on most machines – was to simply tell workers to not have more than one or two Office apps open at any give time – and that included Outlook 2010.

Even more puzzling was the company's approach to the internet connection and bizarre wiring of the building. The cable techs hired primarily to handle store-wiring issues were instructed to fix corporate office network wiring when possible, but even they couldn't make sense of most things. Instead the Help Desk workers had to use toners and probes to trace network jacks to one of several network rooms strewn throughout the complex, and hopefully mark them appropriately for future documentation. The company still needed actual fax machines (analog phone lines), so that made wiring issues at the corporate- and retail-levels even more complicated for the cabling staff. Decades of neglect and half-thought out solutions created a rat's nest of cabling hidden behind most ceiling tiles and walls.

Many offices had to remain vacant until the cabling staff had a chance to actually get the network jack that was on the wall – but dead – operational again. And why did so many offices feature dead network jacks? Because in the past it was easier to pull a line from a nearby wall than to run a new one from the network rooms. As the company grew, the IT staff never bothered to run new wiring in about half of the building.

Another odd thing was that the company was in such poorly organized shape that the business apparently couldn't get credit enough on its own to get AT&T internet at the corporate office. Because of multiple changes in ownership and past technical issues, a corporate account for the corporate office's internet connectivity was not an option, period. The IT management's solution vs. telling the owners to get the issue resolved (properly)? The IT managers would simply get a low-speed AT&T DSL connection in their own name ran to a network closet at the location, and they had that under lock and key. Then, they would use that with a WiFi router they set up to provide owners, managers and visiting people with unfiltered, normal internet access. (You see, most workers were forced to connect to the internet via a partner company's network, and they filtered most web content, for security reasons) Of course everyone and their brother wanted wireless access so that they could visit YouTube and Facebook whenever, and you can imagine the bottleneck that caused on a consumer-plan DSL connection.

The most “brilliant” worker of all was the company's graphic designer (who was a friend of the owners/upper management) ... and his brilliant idea was to use Dropbox to store ... well, everything. When the IT Manager who setup the DSL account discovered the bandwidth was being devoured, he quickly determined who was the culprit – the company's lone day-to-day Mac user. When confronted, the designer explained his workflow, and how he constantly downloaded and uploaded hundreds of MBs of Photoshop files to and from cloud storage on a daily basis, vs. saving them locally on his brand-new, company-provided $2,000+ iMac or on his external 1TB USB drive. It should be noted this worker also returned a personal $3,000 Powerbook to Apple (that he bought with his grossly-overpaid-for-the-area $50,000 salary) because it got a small nick on the aluminum body, due to him putting it in a case a certain way. He's just one example of the weird tech behavior exhibited by employees there.

He was told to not use Dropbox after his workflow habits were discovered, but he never did. Due to cronyism, he was protected. Everyone else just simply accepted that the wireless connection was then bad.

Along with those issues, workers constantly fought with ancient .PST files that stored years of old email messages, and because of that constant Outlook-related support was needed. This was long after everyone had been switched over to Office 365, and those old local .PST files were even described as an acceptable way of managing email. Workers were warned that old email storage choices wouldn't be around forever, so they should migrate what was needed to their new Office 365 mailboxes ... but the vast majority didn't listen.

It's also worth mentioning that the company head honchos alerted the IT management to a crisis one day, after the Facilities Dept. management told them of a failing refrigeration-related issue. Apparently, the large chain of stores had food freezers that were controlled locally (at the store level) by ancient DOS-based PCs from the 90s. One of the store's monitoring PCs had croaked, and suddenly it was IT's task to find a Windows 98-era PC from the basement, that had a specific type of card slot in it (it wasn't even the standard PCI slot that was needed). Not only could they not locate a 15+ year old working PC in their vault of ancient technology, they couldn't even locate a Windows 98 restore disc. And they were given no time or money to get something online, obviously.

Nothing in IT for that business ever worked-, was planned- or was executed-properly, and it made perfect sense that its basement full of giant boxes of cables, towers, crusty keyboards and dead CRT displays was a real-life representation of that company's technology past, present and future.

INFORMATION TECHNOLOGY
CAUTIONARY TALES