THOMAS A. LIMONCELLI

Hey! I have to install and maintain this crap too, ya know!

Thomas A. Limoncelli has written or co-written four books, including *Time Management for System Administrators* (O'Reilly) and *The Practice of System and Network Administration* (Addison-Wesley). He is a system administrator at Google in NYC and blogs at http://EverythingSysadmin.com.

tal+usenix@everythingsysadmin.com

A SURE SIGN THAT SYSADMINS ARE misunderstood and undervalued is that many otherwise great products are difficult to install, maintain, or troubleshoot. Any sysadmin can tell if the installation process was designed as an afterthought. Any sysadmin can point to a variety of . . . I'll be polite and say "design decisions" that make a product difficult to install or completely and utterly impossible to troubleshoot.

A person purchasing a product is focused on the features and benefits and the salesperson is focused on closing the deal. If the topic of installation does come up, a user thinks, "Who cares! My sysadmin will install it for me!" as if such services are free. Ironically, it is the same non-technical executive who dismisses installation and upkeep as if they are "free" who might complain that IT costs are too high and go on a quest to kill IT spending. But I digress.

Installation Woes

I can understand why a product might be difficult to install. It is hard enough to write software, and with the shortage of software developers it seems perfectly reasonable that the installation script becomes an afterthought, possibly given to a low-ranking developer. The person purchasing the product usually requires certain features, and ease of installation is not a consideration during the procurement process. However, my ability to install a product affects my willingness to purchase more of the product.

At a previous job, we were able to massively deploy SGIs because their IRIX operating system installation could be automated. This made it a no-brainer to encourage use of SGIs. When SGI announced they were moving to the Windows operating system, our first question was whether our fully automated Windows installation system [1] could be adapted to their new hardware. We were told in no uncertain terms that this would not be possible for technical reasons related to their custom firmware. We never purchased any of those machines. While SGI's collapse can't be attributed to this one misstep, it did seem to be a symptom of a company that was losing touch with its customers.

Maintenance

Ongoing maintenance and upkeep have similar issues. There have been misguided attempts at making UNIX system administration easier by adding GUIs. A GUI is not automatically easier than command-line tools. Some GUIs get in the way. IBM famously layered a complicated set of commands on top of AIX and layered a complicated GUI on top of that. However, they did two things right. First, their "please wait" icon was adorable. Second, when selecting an action one could always press a function key to reveal the shell command line that was about to be executed. If you had to make the same change on 1,000 machines you did not have to mouse through the same clicks 1,000 times. You simply revealed the command and wrote a shell script to run that command on each machine. Much better.

Although I have not directly used ZFS, I am in awe of the attention paid to making the command line so simple [2]. It takes many times more effort to make a command do the right thing all the time than to simply add more options that an experienced sysadmin will know when to use. Similarly, anyone can add a new button to a GUI, but it takes serious investment of resources to improve the system so that the new button isn't needed.

Perfect Products

I've talked with product managers about why their product is the speedbump that slows me down when troubleshooting a problem that is buried in a network of 150 devices from 15 different companies. In the old days vendors would tell us, "That's why you should buy everything from one vendor—us!" In today's multi-platform arena we're told, "Our goal is to make our product so easy to use you don't need to debug it."

I'm sure that last sentence made you cringe. You get it.

Even a bug-free product requires the ability to troubleshoot problems, because the problems may not be directly related to that product. Imagine an Ethernet switch that is operating perfectly but a user's workstation is not seeing any network connectivity. The ability for a sysadmin to be able to check the status of the connection and verify settings is important in troubleshooting problems like this. Why would anyone make this task difficult? Ah yes, I remember what the product manager said. If the product is perfect it doesn't need troubleshooting.

I've explained to product managers that GUIs are bad when they prevent the basic principles of system administration: change management, automated auditing, backups, and unfettered troubleshooting. We have practices and methodologies we need to implement! Don't get in our way!

The more enlightened product managers understand that the easier it is to automate the installation of their product, the easier it is for me to buy a lot of their product. The more enlightened product managers understand that an ASCII configuration file can be checked in to Subversion, audited by a Perl script, or even generated automagically from a makefile. Sadly, those product managers are rare.

One would think that companies would be investing millions of dollars in research to make sure their products are beloved by sysadmins. This, however, can become a fool's errand.

It costs a lot of money to add features to make a product exceptionally easy to install, maintain, and troubleshoot. In fact, these features may be more difficult than features of the product itself. There are more edge cases and strange situations one must plan for. One may have a good relationship with the direct users of the software, but with the sysadmin organization hidden behind them? That's a lot to ask for. A product manager is expected to know a lot about a product and the industry that uses the product, but knowledge of sysadmin change management, auditing, backups, and such? How can we expect them to know such things when the sysadmin community finds itself at a loss for common terminology and design patterns?

A company could go out of business spending time on these features with very little payback. Enabling software so that a sysadmin can maintain zillions of installations requires specialized expertise, which is expensive to acquire. Scaling software to meet the needs of a single large customer has little payback, especially when such effort could go toward features that attract new customers with less elephantine needs. Spending resources there while your competitor spends money on slick color schemes and spinning icons leads to bankruptcy. In the security world this results in a marketplace where shoddy products are common and the truly great products can't get started [3].

Such features do pay off when customers pay attention to the total cost of ownership (TCO), especially as part of the purchase process. A product that saves money in one area but costs more in maintenance is soon detected when TCO is the focus. The truth is that for most products, operations are more expensive than acquisition. We buy a product once, but it runs hundreds, if not millions, of times. The cost of a hard disk is 20 percent of the cost of providing storage. The remaining 80 percent is consumed by controllers, backup systems, backup media, and other often hidden costs [4].

These operations would be less expensive if product houses could rely on a couple of basic rules of thumb or design patterns to carry them through the process. Shrink-wrapped software already benefits from this: By using common installers they leverage years of experience in getting installation right. As a bonus, commercial software installer kits have APIs to permit automated installs. Open source systems benefit from the user of Autoconf [5] and similar systems.

The solution is more research. More published research will result in a broader array of solutions. It enlarges our toolbox. It makes the world a better place.

I like to think that somewhere out there is a group of researchers studying this kind of thing. I imagine that they find sysadmins who volunteer to be videotaped as they do their job. I imagine the researchers (or their graduate students) poring over those tapes as they try to understand our strange ways. I imagine Dian Fossey studying not Gorillas in the Mist but Sysadmins at the Keyboard.

These researchers do exist.

I've seen them.

For the past two years they've met and exchanged ideas at a conference called CHIMIT (Computer-Human Interaction for Management of IT).

Some of them actually videotape sysadmins and examine what is it about products that makes our jobs more difficult and what makes them better.

My favorite moment was watching a researcher describing his observation of a sysadmin during the heat of a real outage. The sysadmin closed the firewall's GUI and connected to the command-line interface twice, each time in a different window. In one the sysadmin kept repeating a command to output some debugging information. In the other he typed commands to fix the problems. This was something the GUI would never have let him do without risking carpel tunnel syndrome. The researcher beamed as he explained the paradigm we were witnessing. He sounded like he had been lucky enough to catch the Loch Ness Monster on film, but what he had captured was something more valuable: photographic evidence of why sysadmins hate GUIs!

The person sitting next to me sighed and said, “Oh my god. Is that why nobody uses the GUI we spend millions to develop?” I nodded and smiled. The other sysadmins in the audience did too.

I love this conference.

These researchers study people like me and it makes the world a better place.

More than researchers attend. Sysadmins make up a large part of the audience.

The organizers point out that the conference is “an emerging area intersecting the practice and science of systems management, human computer interaction, and service sciences.” They welcome participation from all these diverse fields.

This year CHIMIT will be in Baltimore, MD, November 7–9, immediately following LISA ’09, which by an amazing coincidence is also in Baltimore, MD, on November 1–6.

Mark the dates on your calendar. See http://www.chimit09.org (and, of course, http://www.usenix.org/lisa09) for more information.

Will you be there? I know I will.

REFERENCES

[1] Fullmer and Levine, “AutoInstall for NT: Complete NT Installation Over the Network”: http://www.usenix.org/publications/library/proceedings/lisa-nt98/fulmer.html.

[2] Sun Microsystems, “Solaris ZFS: Simplified Administration”: http://www.sun.com/software/solaris/ds/zfs.jsp#4.

[3] Schneier, B., “A Security Market for Lemons,” April 2007: http://www.schneier.com/blog/archives/2007/04/a_security_mark.html.

[4] Thomas A. Limoncelli, Christina J. Hogan, Strata R. Chalup, *The Practice of System and Network Administration*, 2nd edition (Addison-Wesley Professional, 2007), Chapter 25.

[5] GNU Project, Free Software Foundation, Autoconf: http://www.gnu.org/software/autoconf/.