

Technical Leadership Is Something We Can All Do

TOM LIMONCELLI



Tom is an internationally recognized author, speaker, and system administrator. His best known books include

Time Management for System Administrators (O'Reilly) and *The Practice of System and Network Administration* (Addison-Wesley). In 2005 he received the SAGE Outstanding Achievement Award. He is an SRE at StackExchange.com. www.EverythingSysadmin.com is his blog. tal@everythingsysadmin.com

You don't have to be a team lead or manager to demonstrate leadership. Everyone on a sysadmin team can and should be a technical leader. To me, the children's game "Follow the Leader" exhibits the two essential qualities that are required to be a technical leader.

"Follow the leader" is a game that young children play. One child is selected to be "the leader" and walks as everyone else follows the leader in a single file line.

The leader might walk around the yard or playground, under a swing, between two big rocks. Everyone follows. If the leader takes a big leap over a big rock, everyone else leaps over it the same way. If the leader hops on one foot across a patio, everyone else hops on one foot across the patio. If there is a low-hanging branch obstructing the path, the leader lifts the branch up and walks under, then hands it to the next person, who hands it to the person behind them, and so on.

The leader is doing two essential things. These two things are the most essential parts of being a leader.

1. Go first.
2. Make it easy for others to follow.

They go first. There they are at the front of the line. The game doesn't work if they aren't.

They make it easy for others to follow. When coming to the low-hanging branch they might crawl under it, brush up against it, or lift it out of the way. The leader decides to lift it out of the way. By demonstrating how it is done, the leader makes it easy for others to follow. By making it easy, others can and do follow.

This is a powerful lesson about leadership. If you want to lead, you can't just "encourage" others to do things, you have to do that thing. You have to show that it is possible through action. If you want to lead, you have to make it easy for others to follow. You have to provide the training, the knowledge. You have to clear the path.

If you only do one of those things and not the other, things fall apart. If you aren't willing to go first, people will not follow. If you don't make it easy for others to follow, they'll do something else that is easier instead.

When I was a little boy I was told it was polite to let guests go first. That was true for serving food, but nobody told me it wasn't true for everything else. Trying to lead without going first is a disaster. When leading a group of guests through the house, I'd let the guests go through a door first. Now they're in front and don't know where to go and I'm trying to catch up. It is chaos. The leader must go first, even through a door, to keep leading the group.

Sometimes we forget to make it easy to follow. "I was willing to do it the hard way, shouldn't others?" In a perfect world everyone would be as passionate about an issue as you are, but the truth is that they aren't. We get discouraged because other people aren't stepping up like you did. The truth is that if others aren't following, we haven't taken the time to understand the obstacles they see and help to eliminate them. Often the biggest obstacle is "I don't know how."

Getting five people to show up for a protest is easy. You plus four others who are equally passionate will show up.

What if you want 100 people to show up? It must be considerably easier for people who aren't as passionate to show up. You have to make it easy enough for people who aren't "passionate" but are just "concerned" to show up. They might need more detailed directions how to get there, where to park, how much walking will be involved. All those things are obvious to you but not to the larger group.

What if you want 1,000 people to show up? You still need to appeal to those who are passionate and concerned, but at this level you also have to make it easy to attend for someone who is just "curious." At this level, you need to have a celebrity or entertainer. That would give people two reasons to show up.

That's leadership. What, then, is technical leadership? Technical leadership is when someone paves the way to do things differently than they've been done before. It is when a person makes innovation happen.

Don't confuse technical leadership with "management." Management is a position on an org chart that specific people fulfill. Management is about setting priorities, providing resources, and removing roadblocks.

Technical leadership is different because it is something we can all do; in fact, it is something we all must do if our IT department is to be successful.

Technical leadership is about going first and making it easy for others to follow.

Technology is constantly changing and, therefore, an IT department must constantly be trying new things. For that to happen, someone must go first. Someone must be the first to try something. To transition to the new technology successfully, others must adopt it, too. To get others to adopt the new technology, you have to make it easy for people to adopt it who aren't as passionate as you.

For some of us, trying new things is easy. We're always trying new things!

Sometimes the problem is that there are too many things to try. Which to try first? We have to be selective. What are the three biggest problems in this department? What keeps you up at night? What does our boss complain about the most? Are there new products or services that will fix or alleviate those problems?

On the other hand, trying new things in an IT department is often a luxury. We're too busy to take a day to go through all the trouble to get the thing, learn the thing, evaluate it against other things, and demonstrate that this thing would be worth adopt-

ing in your department; however, if we invest the time required to try a new thing, we can save time for everyone else by sharing what we learned. More importantly, we can make it easy for others to follow in our footsteps. We do this by doing the groundwork, setting up the basic system, and finally creating a way that makes it easy for others to build on it.

Suppose your team is burdened with manually configuring machines. There are automated solutions out there but nobody has time to evaluate them all. Each evaluation means learning the system, deciding how it would fit into your environment, and so on. You take the time to evaluate a few, pick one, and set it up. Maybe it only maintains the configuration on three machines, and the configuration that it controls is modest: just keeping a few files in /etc properly configured. After you've done the hard work, it is time to make it easy for others to follow in your footsteps: you create a wiki page that explains how they can add new machines or start controlling additional aspects of the system. That's technical leadership.

Similar projects:

- ◆ Creating a repository for sysadmin scripts instead of having them scattered in people's home directories. You document how to add new scripts, update existing ones, and replicate the scripts to a new machine.
- ◆ Adopting a request ticket system instead of using email. You set it up with some basic categories and document for the rest of the team how to add/change/delete categories, FAQs, and get the most out of the system.
- ◆ Having a wiki for the team. You give a "brown bag" talk during lunch to teach people how to use it.
- ◆ Setting up a monitoring system so that you know what is broken before your users notice. Setting it up is difficult. Once that is done, configuring it to monitor new machines or services is easy. You document basic add/change processes with clear examples.

Technical leaders don't ask for permission. They may ask for feedback. They may ask for suggestions; however, people tend to dislike change. If you were to propose any of the above projects and ask, "Should I do it?" you'll probably be given 100 reasons why you shouldn't. On the other hand, if you do any of those projects and then give a demo about how it saves them time or improves their life, they will adopt it (especially if you've provided really good documentation: how to get started, how to add/change/delete items in the system, and so on).

The old-fashioned way to make yourself powerful within a company was to hoard information. You are the only person who knows how to do something, and if someone wants it done they must come to you and ask for your good graces. You hide information so that you can control it. You are the great and powerful

Technical Leadership Is Something We Can All Do

wizard that everyone must respect. That is the old way. The new way is to gain power by giving away information. You are the technical leader who set up the new repository, ticket system, wiki, or monitoring system. You went first and made it easy for others to follow. Now your power comes from teaching others to use that system. You are powerful because everyone in the organization remembers that you were the person who taught them how to do that thing and the other thing. You are powerful because your influence extends throughout the company because of all the people you've helped.

Technical leadership is something we can all do. For a modern company to survive, technical leadership is something we all must do. In fact, for the greater system administration community to survive we all must be technical leaders.

Professors, Campus Staff, and Students— do you have a USENIX Representative on your campus? If not, USENIX is interested in having one!

The USENIX Campus Rep Program is a network of representatives at campuses around the world who provide Association information to students, and encourage student involvement in USENIX. This is a volunteer program, for which USENIX is always looking for academics to participate. The program is designed for faculty who directly interact with students. We fund one representative from a campus at a time. In return for service as a campus representative, we offer a complimentary membership and other benefits.

A campus rep's responsibilities include:

- Maintaining a library (online and in print) of USENIX publications at your university for student use
- Providing students who wish to join USENIX with information and applications
- Distributing calls for papers and upcoming event brochures, and re-distributing informational emails from USENIX
- Helping students to submit research papers to relevant USENIX conferences
- Encouraging students to apply for travel grants to conferences
- Providing USENIX with feedback and suggestions on how the organization can better serve students

In return for being our "eyes and ears" on campus, representatives receive a complimentary membership in USENIX with all membership benefits (except voting rights), and a free conference registration once a year (after one full year of service as a campus rep).

To qualify as a campus representative, you must:

- Be full-time faculty or staff at a four year accredited university
- Have been a dues-paying member of USENIX for at least one full year in the past

For more information about our Student Programs, contact Julie Miller, Marketing Communications Manager, julie@usenix.org

www.usenix.org/students