

Memoirs of a Self-Loathing IT Professional

By Bernie Wieser

© 2014

Smackdown

The last challenge Anne gave me wasn't much of a challenge though it did have extra responsibility. She assigned me the end-to-end upgrade for Banana Energy Corporation's contract management system. It was one of those funny systems. Usually IT had several environments that allowed for development, testing, training, and production - but not this puppy. We didn't have much of a relationship with the vendor and the software contract had no provisions for multiple environments. IT leadership only acquired one license because it was too expensive to get licenses for each environment. This meant we couldn't really test the system upgrade before going live.

I made sure that we had a back-up and back-out plan. I arranged that the upgrade would be performed on the weekend so the ten or so power users wouldn't be affected. IT shops do this a lot – try to schedule outages after hours or on weekends so as not to annoy or interfere with clients too much. We don't get overtime for these events as consultants because we bill by the hour. Also, if something goes terribly wrong there are fewer people around to see the shit show. Since I started I'd witnessed several upgrades go into full panic mode over a weekend. The clients never knew the hardships endured when they arrived at their desks Monday morning. My upgrade went without a hitch. I was only distracted from Saturday morning gaming for a couple hours.

I had full access to the contract system and like a good citizen I didn't go snooping to see who was making how much or what Banana had paid for this or that. I was only a little curious but I fought the temptation to look. Sometimes administrator access can be like a drug that induces a power trip. I knew one system administrator at university who would regularly read other staff and students' email. No one would have known except for when he discovered his girlfriend, who was a grad student, was cheating on him. He discovered this by reading her email and promptly deleting all the files that she and the other offending party had created. The other party, a professor, was slightly annoyed at losing his years of research and was successful in having said admin escorted out of a job and off campus by police.

It seemed Anne had forgiven me for talking to the executives about how to do things better. We didn't talk about it afterwards. There didn't appear to be any negative fallout or for that matter any changes whatsoever. The machinery hummed along like it had before. Anne gave me full stewardship of the contract system which made it a continuous assignment. And right-sizing in her group found me becoming the power user "steward" of ten more applications. I supported the users, administered the systems, and kept the systems' lights on. This support was boring because there was no coding involved. I suspected some of the systems weren't even used because I never had a support call on a couple of them, nor did I know who the users were.

After a few months Anne's group was reorganized to include another group and we had a few new faces. These new folks supported oil and gas marketing, the area responsible for the buying and selling of product. I was asked to job shadow one of their guys and he was asked to shadow me. His name was Alan Chang, and unlike many of us who were consultants, he was an employee. He treated his job very differently from the rest of us. Work got done when it got done. Corporate social events took priority over deadlines and Alan participated in all of them. Alan wasn't the sharpest tool in the shed when it came to making software. He was a geologist by training. In our personal meetings he explained that his path took him on a journey to IT management – writing software was just a transitional step. That scared me a little because when I spoke with him he didn't really seem to understand what he was doing. I wondered, *How can anyone who doesn't get programming, or care, manage people that do?*

I had to read some documentation Alan wrote in order to support his applications as his backup. In one document he kept referring to *pup-up menus*. It took me a moment to realize he was talking about *pop-up menus* and was writing phonetically. His glossary was also funny because there were definitions for words that weren't relevant to the application, some definitions were completely wrong, and others used jargon that would only mean something to an IT person.

Application	Computer codes to implement functions for users.
Prod	The system where the users should run the application.
TQA	The Quality Application where users test.
User	A person that uses the application.
Web Browser	The program used to browse the inter-web.
Web Server	The system that sends the software to the user for running.

When studying Human-Computer Interfaces in school we were taught the only thing worse than documentation was bad documentation. It's hard enough to get clients to read documentation, but make it unclear and inaccurate and it's a guarantee of wasted effort in producing a paperweight. I wouldn't have explained what *servers*, *browsers*, *applications* or *users* were because I'd assume a person working in a corporate setting would already know. I'd have a single note to explain that *production* and *total quality assurance* were technical environments where *servers* hosted *applications* that users could run through their *web browsers*. I'd let the users know how to figure out the difference between test and production because on more than one occasion I had a support call where a user had misplaced their data and so the system was obviously broken. *I can see it but it doesn't appear in my report* was a clue.

So it came as no surprise that one of Alan's applications on its latest update was causing trouble. "I can't figure it out," he'd tell Anne, "it works fine on my computer and in DEV but when I promote it to TQA it just chokes." Anne called on me to see what I could do because *I was good at fixing those things*. The application was web based and the web servers were administered by a different group. When I checked the deployment package everything was set up the way it should have been. Alan hooked me up with an administrator named Paul Barton to help because we developers didn't have full access to the test environment. I recognized Paul as someone who rode the same train to work at the same time that I did.

Paul seemed nice enough when I first met him. He explained he had a quick look at the errors but he couldn't figure it out. I asked him to send me the logs so I could see them myself and he obliged. The error implied that some function was being called with more parameters than it allowed. That was odd because everything linked up fine in the package. I asked what the username and password were for the development machine and Paul told me: *DEVSERVER* and *DEVPASSWORD*. I asked Paul to send me the configuration from both environments and everything seemed configured the same. In passing conversation I discovered Paul was also an avid gamer (that explained the dark circles under his eyes.)

I added some diagnostics to give me more details and this information was dumped to the *console* so I needed to see it running in the test environment directly. Paul had the figurative keys to this test system. As I needed to watch over Paul's shoulder I tried to book time with him but Paul was a very busy man. I had trouble booking his calendar as it always seemed full and he had stopped responding to my email. The client in marketing was getting increasingly frustrated their new functionality hadn't been deployed. I complained to Jeff about it and he told me *just log in to TQA as TQASERVER and use TQAPASSWORD*. I marvelled at the simplicity and insecurity of it; using a slightly modified version of DEV credentials. I hoped the same pattern wasn't used for production systems but resisted the urge to try.

I remotely logged in with the credentials. It was a UNIX system like we used in school so I was intimately familiar with it. I entered a command and saw Paul was online too. I proceeded to install the app with my diagnostics and started it up, watching the configuration information dump out to my monitor. I saw the locations of the libraries were not from the application's package but from the web server itself. This was a *no-no* and it meant the web server was configured incorrectly. I confirmed this with Jeff who told me it was against our internal procedure to rig things this way but sometimes the admins went against their own advice as a quick fix for some issues. Sometimes a developer forgot to put everything needed in an application's package and those developers were likely no longer at Banana. Frequently developers were project based and when the project was over they went away taking all their knowledge with them. This left people like the admins to clean up unexpected, undetected messes.

Being in the thick of it I sent Paul a message through the system.

“Hi Paul, it’s Mark. I think I found the problem and can fix it if you want, or I can tell you how to fix it,” I wrote. I waited a few minutes for a response. *Maybe I should call?* I thought. Then I wrote again, *“You just need to delete this library from the web server’s path and everything should work.”* Another few minutes went by and then I got a response I didn’t expect.

“How did you get on here?” Paul wrote.

I was perplexed. *Am I not allowed to be on here?* I thought. I assumed that if Jeff knew the password to this system that he would have permission to use it. Jeff could use it, why couldn’t I? I decided to play dumb. *“I had the test username and password,”* I typed.

“Who gave it to you?” Paul wrote.

“I don’t remember,” I fibbed. I didn’t want to give up Jeff because he always helped me.

“You shouldn’t be here,” Paul replied. *“I am going to write this up as a security access violation and change the password.”*

I was stunned. *“Paul, I found out why Alan’s app wasn’t running and can fix the issue,”* I typed.

My console then told me I was disconnected from the server. *What the hell?* I thought. Paul, as super user, had terminated my connection. I immediately looked up his phone number on the company white pages and called him. He didn’t pick up the phone. So I wrote a long email to him, Alan, and cc’d Anne explaining what was broken and how to fix it. The only person to reply was Anne, thanking me for finding the issue. Sure enough within a few hours the problem was fixed.

A couple days later I saw Paul on the train. It was the busy rush hour and I could clearly see him. He completely ignored me. I thought of pushing my way through to ask him what his

problem was and why he would tell on me for trying to help. I could see he was uncomfortable. I think I glared at him the entire trip. Never once did he make eye contact, though he did try to maneuver himself such that other passengers were obscuring the line of sight. Finally our stop was coming up so he would have to get off and out the same door as me. To my surprise he didn't move. *He would rather miss his stop than face me?* I thought. Sure enough he skipped our stop.

Anne's morning status meeting with the group went well. After Ted left no one bothered to send in their status reports and Anne never asked for them. This validated our beliefs that Ted was a micro-managing control freak. Anne trusted us to stay on top of our own work and used the meetings to make sure everyone knew what everyone else was doing. The former marketing IT guys now in attendance didn't have much to say in our meetings. Jeff told me he suspected they were just waiting for their next reorg. "They get juggled a lot," he said. "No one wants to own them."

Near the end of the meeting Anne brought up the incident. "I had a meeting with IT Security about a possible breach," she said. "There was an incident report filed about one of the developers gaining inappropriate access to one of the test servers. Now, I told the auditor that this was not inappropriate because I *asked* the developer in question to fix a problem. He reminded me of our access policy, segregation of duty, and requested I remind you guys of the policy. I understand it can be frustrating when our clients are asking for things and we have to wait for the systems guys to get on it. The systems guys were just doing their job trying to make sure the controls are in place. In future I ask that you guys do not communicate with the systems guys directly but instead go through me. I can light the appropriate fire under their bosses' butts to get them engaged. We got off with a warning this time for breaking the rules. Is that okay with everyone?"

"Yay bureaucracy," Dean scoffed. "Not exactly efficient is it?" (Dean had recently, regularly been wearing a suit to work. When we asked him about it he just said he was dressing for success.)

Anne didn't answer him directly. She leered slightly over the rims of her glasses.

I couldn't help myself. "I don't understand why we have administrative access to some systems in production but we have no access for others in test."

"We own those production systems Mark," Anne replied. "We don't own the UNIX systems or the web servers that run on them."

"The contract I signed tells me what I can and can't do; right and wrong. I'm not sure how logging into the test environment to watch a diagnostic dump to console is a security incident," I replied. "I also don't get why the auditors have to be involved."

"It is what it is," she said shaking her head. "Access control is in the IT security policy."

"Auditors are the big stick," Dean stated. "Audit is a dirty word."

"You know what I do in my apps?" Jeff started. "I just have them email me the diagnostic output when something goes wrong. That way I don't have to rely on anyone."

Not a bad idea, I thought. "I'll do that in the future but I don't think it would work in this case. I needed to see the web server start up."

"That's in a different log," Dean said. "*You* don't ever need access to the box, you just need the guy with access to open up the log under file sharing. I've asked for this a hundred times. They won't do it, and won't tell me why not."

"You can do that?" Alan asked surprised.

"Okay you can save your technical discussion for afterwards," interrupted Anne. "Just remember use your discretion when talking to other groups. You're all doing a great job keeping our clients happy. The marketing group was very happy we solved their issue and *they* are our client. Keep the client happy. Leave the systems guys to me."

"Forgive me," Dean started with his typical sarcastic tone, "but these systems guys are super technical and shouldn't they be talking to technical people, like developers?" Dean didn't

have much respect for Anne. Her background wasn't IT, it was accounting, and Dean had offered up veiled questions about her competency before.

"I *will* take care of it," Anne responded with a decisive finality.

The meeting wrapped up and Jeff followed me to my office.

"You know the UNIX guys are a different breed," he said. "You just showed Paul up because he should have known better. They're introverted, solitary gurus and they don't like change. They'd be happy if the clients would just go away so they can rule their kingdom their way."

"I'm starting to see that," I replied, "but Paul didn't have to snitch on me. I was just doing what I was asked."

"I know. It's okay," Jeff said as he grinned through his straggly beard. "Let them deal with their daemons and transcendental functions." He paused. "And I'm not pissed off at you for losing access to test either."

I hadn't thought of that. I think Jeff was thanking me for not turning him in. "How did you come to get the password?" I asked.

"C'mon, I guessed!" he laughed.

"Seriously?" I replied.

"No, not really," he said. "The systems guys have all their passwords documented and published for anyone to see. It's for recovery purposes and it's their policy. You can find them by doing a search in the corporate document management system – you don't even need to open the documents. I found them just doing a search for *maintaining test web servers*. They're pretty good about keeping that stuff updated. If I need the credentials again, hypothetically speaking of course, I'll just go look them up."

On the ride home that day I wondered if there were some lessons for me. *The solutions guys and systems guys shouldn't talk?* It seemed wrong. But there were a few things that stuck in my brain from that day forward. We IT people are not all on the same team. The client doesn't always come first. Fixing things isn't as important as following the rules, unless of course you make the rules, and then you can break them as much as you want.