BUZZ CHRONICLES > TECH
Saved by @jay_millerjay
See On Twitter

Twitter Thread by Zed A. Shaw, Writer

Zed A. Shaw, Writer
@lzsthw

When you complain about an open source project's management you'll get the reply, "Use something else if you don't like it!" That'd be true except for the tiny problem that *many* projects also strive for complete monopolistic dominance.

Let me tell you about Java in early 2k:

I was programming in Java since it's release, and all through the late 90s into the early 2000s. At first Java was very determined to unseat C++ and it used various propaganda tactics to turn C++ and malloc into a weird enemy for all of us to hate. This worked great!

Everything was fine though because there were plenty of jobs in tech because of the dot-com boom, and then suddenly there were not jobs because of the dot-com bomb.

Overnight it seemed as if the *only* way to get a job was at giant companies, and they *loved* Enterprise Java.

Now, this is the golden ticket for many open source projects and corporations. They all want to be the top dog #1 in the industry for their particular use, and once they gain that position they become abusive by ignoring user needs and extracting wealth from them.

With Java the dominance was so insane that *every* language had to copy Java's weirdness to survive. If you wonder why a language has some odd CamelCase things it's because early Python, Ruby, and others had to copy Java's semantics and style to even be considered "serious".

Then Ruby on Rails breaks on the scene around 2004-2005 took on Java's dominance with the exact same kind of propaganda tactic.

"No more XML situps!"
"Convention of configuration!"

And this worked just the same way Java's did and RoR started eating into Java's lunch.

Right at this transition time from Java to Ruby I got into Ruby and started trying to use it. The backlash from Java developers was INSANE! I caught one logging into machines and sabotaging the RoR processes. Another would yell at me in meetings about Rails. Others quit over it.

Imagine that. You love Java so much that you'll *quit* just because another guy 2 cubicles down is using something not-Java. That's the power of propaganda and the exact position Ruby on Rails would later capture with the same results.

The goal of all projects is capture.

Later Ruby on Rails unseated Java and became the dominant force, and once again if you did anything else you were ridiculed and people were threatened. I started using Python and people had the same reactions, freaking out, quitting, yelling, shaming, everything Java did.

Now, this social pressure is insane and most people can't resist it. I know it definitely cost me jobs trying to switch languages in these situations. Changing languages also cost me friends who started to hate me over just using something else, plus a whole slew of crazy haters.

So, how is it possible for someone to tell you "just use something else if you don't like it" while that same person is also doing their very best to capture the market and make it impossible to use anything else?

It's actually passive aggressive authoritarianism.

The project dominates tech.

When you have no choice but to use their project they use this control to exploit you and ignore your needs.

When you complain about their actions they tell you to just leave.

But, you can't, so they actually mean: "Shut up and do as you're told."

There's also the insane switching costs when learning a new programming language and platform. "Just use something else" takes on a whole meaning when the only way to use something else is to spend a year completely changing your whole platform and language.

Throw into this the practice of inventing shiboleths to gate keep people. You want to switch from Java to Ruby? Welp, better stop using for-loops and use .each or you'll never get a job.

All this adds up to "just use something else" is only an abusive control gesture.

What *should* happen is if a project is so dominant that people have no choice then the project should be run democratically. Banning people from these large projects should also be done carefully because these actions can put people completely out of work over minor things.

For example, the Go project bans people for using words with the letters "ass" *in* them. Someone was banned for rightfully calling something "halfassed", and this ban was done unilaterally with zero oversight by an unelected cop.

That is authoritarianism, but...

In the early days of Go before its dominance they would never have been so sensitive because they wanted to attract as many people as possible. I know many of the early Gophers and many of them are total trasholes who would never survive today's Go policies.

Finally, this is also why you should get *very* good at learning new languages and platforms. The only reason I was able to survive Java, Ruby, and Python's authoritarianism is because I could learn a new language fast and make good things quick, which removed their power.

But, even being able to learn new languages doesn't reduce the risk of some jackass in a dominant project just deciding they don't like you one day and using their power to destroy you. Always have your backup plan in the works and never trust a project will behave ethically.