# Stockholm Syndrome in Software

Adrian Kosmaczewski

2022-12-16

Developers working for a particular vendor tend to develop a bizarre version of Stockholm syndrome[1]. It's something I've witnessed at least twice in my career.

- Windows developers who put up with the worst times of MS technology around the end of the 2000s, migrating[2] to Ruby on Rails, the Mac, or wherever possible.
- Apple iOS and macOS developers putting up with buggy, incomplete, undocumented Swift, SwiftUI, and Xcode versions. One day I had enough[3], and some time later I migrated[4] somewhere else.

Those developers will not only put up with the atrocious developer experiences crafted by their alma mater companies; they will even defend those horrors in the name of market reach, security, marketing, design, or any other reason. By "defend," I even mean openly attacking those who dare criticize their technology choices.

I quoted Steve Troughton-Smith last week[5] talking precisely about this issue:

> A little? It's deeply unpopular to use ObjC or say you like ObjC over Swift. Swift has a truly massive hype train, and you definitely don't want to stand in the way of it.

This is precisely the point: choosing a technology for a living is challenging and requires a significant investment from developers: courses, training, books, and lots of practice. All of that takes years to complete, so imagine your reaction when you find out your platform vendor is slowly but surely eroding your capacity to deliver quality software in any way.

So developers clutch at whatever straws they can find to justify their decisions and defend their bread-winning skills. Understandable.

I preferred to migrate to other galaxies[6], and I have done that at least twice. I will most probably, do that again. I am not attached to a particular tech, even though it might appear like it's the case.

[1] https://en.wikipedia.org/wiki/Stockholm_syndrome
[2] /blog/migration-the-return/
[3] /blog/courage/
[4] /blog/migrating-from-macos-to-linux/
[5] /blog/what-objective-c-3.0-could-have-been/
[6] /blog/the-developer-guide-to-migrate-across-galaxies/

In the case of people working with (or instead suffering because of) Apple crap, I can only feel pain. For example, every time I see on Mastodon[7] good friends dealing with that joke of IDE that is Xcode (in a complete state of destruction since version 4 circa 2010) or the Swift programming language. A language whose versions 1 and 2, let's admit this, were full alpha versions, absolutely not ready for prime time, and probably the crappiest versions of anything ever delivered by Cupertino.

And to add insult to injury, this week JetBrains announced they would stop producing and supporting AppCode[8]. Really bad news; it was a terrific product, unfortunately it never got enough traction, partially thanks to Apple making it impossible for third-parties to provide viable alternatives to Xcode. I really hope Swift Studio[9] by Marcin Krzyżanowski[10] will have better luck.

Speaking about botched developer experiences, Visual Studio .NET ranks high, particularly in its 2 or 3 first releases. It was a clumsy set of tools that could barely remain running for a few minutes before crashing into oblivion. The alternative to .NET was to go back to COM components in various forms, so yes, I'd instead try to get this ASP.NET app to compile and maybe even run; thankyousomuch.

I've seen too many failed projects based on C# 1.1 or Swift 2 for my taste. Both Microsoft and Apple, 13 years apart, played the same game, using their users as testers[11]. It's a shameful situation happening once and again in an industry that likes to make you hate every second of your breathing life as a developer.

No wonder many developers go back to the command line[12], vim, Emacs, and bash; they all certainly propose a limited experience compared to visual tools, but one that at least is stable, predictable, reproducible, and straightforward. Boring[13], even.

I could say the same about FOSS code in general. There's no marketing involved; you know it's going to crash, eventually, anyway. But at least you didn't have to listen to vain promises beforehand. And you will still be able to compile and run your code 20 years from now. Not a tiny proposition, indeed; try to run that 2014 Swift 1 code or that C# 1.0 code from 2001 nowadays; good luck with that.

Stockholm syndrome in software is directly proportional to the hypocrisy and hubris displayed by vendors on their marketing campaigns. At some point, the coin drops, and developers move on somewhere else. It can take a while, depending on the size of your investment in said technology. Sadly, many good developers leave the industry altogether because of these situations, and that's a net loss for all of us. As I said once[14], we need less evangelization, and much more honestization.

---

[7] /blog/mastodon/

[8] https://blog.jetbrains.com/appcode/2022/12/appcode-2022-3-release-and-end-of-sales-and-support/

[9] https://swiftstudio.app/

[10] https://krzyzanowskim.com/

[11] /blog/users-as-testers/

[12] https://github.com/readme/featured/future-of-the-command-line

[13] /tags/boring-languages/

[14] https://deprogrammaticaipsum.com/less-evangelization-more-honestization/