

sean goedecke

Dangerous advice for software engineers

I'm a big fan of "sharp tools". These are tools that are powerful enough to be hugely helpful or harmful, depending on how they're used. Most forms of direct production access are in this category: like ssh or kubectl, or a read-write prod SQL console. It's also possible to give "dangerous advice". Dangerous advice is dangerous because (like sharp tools) it takes competence and judgment to use well. Giving the wrong person dangerous advice is like giving the wrong person production SQL access - they might go off and do something enormously destructive with it.

I give a lot of dangerous advice on this blog. For instance, I advise readers to:

- [Make your own decisions about what to work on](#)
- [Deliberately break written company rules sometimes](#)
- [Take strong positions even when you're uncertain](#)
- [Identify as a bit of a 'grifter'](#)
- [Deliberately avoid all non-shipping activities](#)

I worry about some readers making the wrong decisions about what to work on, or breaking the rules in the wrong way, and so on. I don't like the idea that I've contributed to harming someone's career. But I think dangerous advice is important to write down *somewhere*.

Most career advice is fake

The main reason why is that **strong engineers crave dangerous advice**, for the same reason they crave sharp tools. Most career advice is fake: written to avoid liability, or to impress people. If you're serious about getting things done, you'll end up doing some of these things because they're obviously helpful.

It can be deeply alienating to *know* things don't work the way they officially should, but to have nobody to talk about it with. When I was a more junior engineer, I really appreciated the few peers I had who were willing to get real with me (off the record).

Managers can't tell it to you

Another reason why is that **managers will almost never give you dangerous advice**, even if it's what you need to hear. If a manager tells you to ignore company policy, and you do it wrong - for instance, if you post in Slack that you're doing it and your manager said it was OK - then that's bad for them. In fact, it's much worse for them than it is for you. Tech company leadership often views engineers as useful idiots. Managers are expected to be professionals.

However, lots of managers wish they could give you advice like this. They certainly appreciate it when you follow it. I've never been a manager, but it must be incredibly frustrating to manage strong engineers who would be much more effective if they approached work a little more tactically (and a little less according to the written job description).

Dangerous advice requires courage to follow. It's very high-risk, high-reward, and is thus disproportionately useful to strong engineers (and harmful to weaker ones). If you don't feel comfortable following it, you definitely shouldn't. But if you're already operating like this sometimes and wondering whether you're making some horrible long-term mistake, I don't think so. Or if you are, I'm making it too!

edit: This post made it to [Hacker News](#), where it got a mix of very positive and very negative responses. It made me wish I'd explicitly included more of the points from my previous post [How strong engineers break the rules and get away with it](#).

The biggest disagreement I have with the negative commenters is that they seem to think organizations *are* their sets of written rules, and that the primary mode of operation in an organization is formal, structured collaboration. I need to write a whole post about this, but I think this is the classic [Seeing Like A State](#) mistake of over-prioritizing *legibility*. All communities have a load-bearing illegible component. The main thrust of this post (and most of this blog) is that you should think carefully about the illegible parts of your own work.

If you liked this post, consider [subscribing](#) to email updates about my new posts, or [sharing it on Hacker News](#).

April 5, 2025 | Tags: [good engineers](#), [tech companies](#)

[subscribe](#) | [about / popular](#) | [rss](#) | [book](#) | [what I'm currently working on](#)

[← How strong engineers break the rules and get away with it](#)

[Software engineering under the spotlight →](#)