



Focus on Family More Than Code

JUNE 2026

8 min read • 1,849 words

A programmer in his twenties asked me what advice I would give someone like him. I said: focus on family more than code.

I think it is the best advice I have. I also think I am close to the last person who earned the right to say it, which is more or less how I know it is true. I spent my own twenties optimizing the wrong system. Very efficiently. With excellent test coverage.

So take what follows as a field report from the far side of a trade I already made, not a sermon from a mountain. I am not your mentor. I am closer to your outcome, the one you are presumably trying to avoid. Here is the short version, and then the long and expensive version of each.

- Focus on family more than code.
- The mania looked like productivity. It always does.
- Marry the person who tells you the truth at cost.
- Do not make load-bearing anything you do not control.
- Warn the people you love before you ship the breaking change.
- Sleep is a load-bearing wall, not a productivity tip.
- Ship to a userbase of one, and make sure the one is happy.
- Treat what you believe now as a snapshot, not the final release.

Now the part where I explain how I learned each one the slow way.

Focus on family more than code

This is the whole essay. The rest is footnotes.

For about a decade I believed, somewhere under the part of me that would have denied it, that the code would eventually be finished. That there was a version where the library was done, the inbox was empty, the thing was stable, and I could finally walk out of the room and go be a person with the people who live in my house. There is no such version. I checked. I checked for about ten years.

The people in the house kept being alive whether or not the build was green. My son is four. There are others here I am not going to put on the internet, which is itself a lesson I learned a little late. The people asleep down the hall outrank every release.

Let me be plain, because this is the part I cannot make funny. I gave years to a download counter that I am not getting back, and the people I gave them away from were the ones I say I love most. [I have written the long version of that reckoning elsewhere](#). The house I live in matters more than the code I ship. I could not have written that sentence at twenty-five. It is the truest one I have now.

The center of gravity has to move. The code is supposed to serve the life. I had it backwards for years: the life kept getting scheduled into the gaps around the code, on the way to something more important, which was always more code.

The mania looked like productivity

Nobody warns you about this one, because the culture that would warn you is the same culture handing out the applause.

Some of the years I was most productive were the years I was getting sick. The all-nighters, the electric certainty, the feeling that I had finally locked in and could see the whole system at once. From the outside, including my own outside, that looked like discipline. The intensity that built [Requests](#) and the intensity

that kept landing me in a hospital were the same intensity. [The engine had two outputs and nobody could tell them apart](#), because from the outside all you can measure is the code, not the cost to the person writing it.

I am not going to tell you how to read your own engine. I cannot reliably read mine even now, which is sort of the whole point. I will only say that "I have been so productive lately" and "something is wrong" are not always different sentences, and the world will only ever clap for the first one. Somewhere a younger version of me is awake for the third night running because the work is flowing beautifully, and he is certain this is the good part. He is not available for comment. He is busy.

Marry the person who tells you the truth at cost

For a long time I assumed the risk in love was the other person. Loving someone too strange, too intense, too much. I had it backwards. The real risk would have been arranging a life where nobody close enough to matter was allowed to tell me the truth.

[My wife sees the weather coming before I feel the wind](#). Not as a romantic flourish. As survival architecture. She notices the small shifts, the sleep slipping, the speech speeding up, days before I will concede that anything is happening at all.

What I want you to hear is the cost, because honesty that costs the teller nothing is cheap and everywhere. Sarah's honesty costs her. At the far end of the worst of it, the truth she has had to tell has not been a quiet conversation across the kitchen; it has been [the kind she has written about in her own words](#), the kind that ends with a hospital, and it takes something out of her every time. That is the part of this I am least able to write about casually, so I will not. I will only say: build the center of your life around someone whose love takes the shape of telling you the truth, and then spend the rest of it making it safe for them to keep paying that price. It is worth more than anyone who keeps you comfortable.

Do not make load-bearing anything you don't control

Every engineer eventually learns not to build their company on top of an API they do not own. The platform changes its terms, deprecates the endpoint, and your house falls down on a Tuesday for reasons that have nothing to do with you.

I learned the same lesson about a self. I had welded my identity to a download counter. The numbers were a scoreboard and I was winning at the only game that had ever let me play, and it felt wonderful, right up until I understood I had made my sense of being a real person depend on a metric maintained by strangers.

My son has no idea his father once believed download counts were a load-bearing part of a personality. I would like to keep it that way as long as I possibly can.

Keep the load-bearing walls things that are actually yours. The people who would still know you with the repository deleted. The work you would make if the counter read zero. [A scoreboard is a fine thing to glance at and a terrible thing to stand on.](#)

Warn them before you ship the breaking change

In software you do not yank the thing people depend on without warning. You ship a deprecation notice. You give people a version or two to adjust, a migration path, time to ask questions. You do this even when the new version is plainly better, because the unannounced change does more damage than the change itself.

People build their lives around your patterns the way apps build on an API. The person you love has wired themselves to the shape of your promises. So when something big is changing, and it will, you say so out loud, early, before it is a done deal. I learned this by getting it wrong: by coming home with decisions already made, about work and money and how I was handling my own head, and presenting them to Sarah as finished. [That essay exists too.](#) It took me an

embarrassingly long time to notice that running a marriage on semantic versioning is, itself, a slightly broken way to live. The metaphor keeps holding, though, which I have decided to find reassuring rather than alarming.

Sleep is a load-bearing wall

This is the least mystical item on the list and the one holding up the most.

I can only tell you what has been true for me, and it does not generalize into medical advice, so please do not take it as any. For me, every crisis I have ever had was preceded by lost sleep. The one that started all of it was four days awake. So somewhere along the way sleep stopped being a wellness tip and [became structural](#). A night traded for output is a withdrawal from a wall that holds the roof up, and I have a roof I would now very much like to keep.

It is, I admit, the least sellable idea I own. Nobody is building a content empire on "go to bed." You cannot optimize it into a personal brand. It just quietly keeps the house standing, which turns out to be the entire job.

Ship to a userbase of one

A breather, because the last few were heavy.

[Requests serves thirty-some million installs a day, and I carried it like a piano on my back](#). The website you are reading this on runs on a framework with exactly one important user, who is me. The best things I have made lately fit one hand because they more or less are my hand. The userbase of one has never once filed a rude issue.

The gospel says build for scale, find your market, capture everyone. Sometimes, sure. But you do not need a market to justify making something, and the thing that fits exactly one person can be the most uncomplicated joy you have. The big number cost me more than it ever paid. The small one just makes me happy, which I spent a long time assuming was not allowed to be the point.

Treat what you believe now as a snapshot

Last one, and it quietly governs all the others, including the eight tidy bullets at the top.

In 2016 I was too sure I was fine. In 2019 I was too sure about what would never work for me. Opposite errors, and the same error: [treating the current snapshot of what I knew as the final release](#). Every confident thing I have ever said about my own life has had a half-life.

Which means this essay has one too. Somewhere in here is a sentence I am too sure about, and the genuinely humbling part is that I cannot tell you which. I will find it in a few years, the way you find a bug you would have sworn was not there. Write down what you believe anyway. Commit it under your own name, not because you are right, but because the corrections are the treasure, and you cannot correct what you never wrote down. The people who will one day fork your defaults deserve to read what you actually meant.

I gave a stranger in his twenties the best advice I have. The only reason I had it to give is that I spent ten years not taking it.