



The Reality of Developer Burnout

2017

5 min read • 1,093 words

Themes: Mental Health



Burnout is, unfortunately, a very real phenomenon in software development — especially when creating and maintaining open source projects with large numbers of users.

Research by GitHub and others shows that maintainer burnout is endemic in open source, with many developers reporting feelings of obligation, stress, and exhaustion from unpaid labor supporting thousands of users while managing demanding day jobs.

I've experienced it, and I wanted to share my personal experience with the subject.

It happens to everyone that writes code all day long — the sudden feeling of "I'd rather do anything else than this right now" — even though writing software is one of your favorite activities in the world.

You suddenly realize that you've been eating ice cream for three meals every day for years on end. You're tired of it; you don't want to see ice cream any more. People who eat ice cream occasionally won't understand this; how could you possibly want less ice cream?!

— Gary Bernhardt

I have some personal experience with software development burnout, and some tips for recognizing it, avoiding it, and simply dealing with it.

The Inevitable Entropy of Goals

After setting some open source community goals for myself in 2011, I went hard to work on them, and, with some luck, I was very successful (with the Requests project, in particular). As its popularity rose and rose, my drive to continue to create new projects (datetime for humans, for example), fell. All while the burden of supporting the needs of the massive userbases of my successful projects and the pressure of maintaining those projects grew.

This is what I wanted, right?

Burnout is sneaky. It doesn't usually announce itself. It slowly grinds at you until these feelings become the new normal, and at that point it's not easy to dig yourself out of the hole.

— Zach Holman

The 410 GONE Situation

I remember a moment, while dealing with a newly formed chronic migraine (NDPH), evaluating my life, lying on my couch, where I sit right now. I was having thoughts, like tweets, about some of the political issues in the software

world at the time, and I found myself quite stressed out about these things, which was a problem in and of itself. Not only that, but I found myself censoring my own private thought processes, in the silent comfort of my own home, because of the public opinion of people I follow on Twitter.

Once I recognized this, I realized there was a big problem, and it had to stop immediately. My first thought was to pull what is known as a 410 GONE situation. Of course, I did not do this—but, I considered it heavily.

410 GONE is the infamous move that Mark Pilgrim, Python Developer and human, made for unknown reasons to distance himself from the developer community, and I assume, retain his own sense of identity while feeling overwhelmed by the pressure of being a "leader" in open source.

Mark Pilgrim's sudden disappearance in 2011—deleting his website, code repositories, and online presence—became legendary in developer circles as an extreme response to internet fame and the pressures of open source maintenance. The HTTP 410 status code means 'Gone' and indicates a resource has been permanently removed.

He, suddenly, deleted all of his public code from the internet entirely, leaving all of his users to re-host unofficial remnants of his useful legacy.

This is, in my opinion, both the epitome and worst case scenario of the burnout cycle. I was very close to deleting all of my projects from GitHub, all my slides from SpeakerDeck, destroying my website, and moving on with my life in peace.

But, I didn't. Why? Because those are some of the most important things in the world to me. Why would I destroy something I worked so hard to create, and am so proud of?

Publish-Only Mode

So, instead, I decided to fix the root of the problem. I realized that I was letting too many people into my world, not delegating enough, and needed help maintaining my projects. I didn't want to lose what I valued most about my position within our community—being able to influence the world I cared so much about.

So, I unfollowed everyone on Twitter. Every single person. I stopped paying attention to tech trends and reading hacker news. I went into publish-only mode.

This strategy reflects research on information overload and attention residue—the way constant context switching between social media and work depletes cognitive resources. 'Digital minimalism' advocates like Cal Newport suggest similar approaches for maintaining focus and mental health.

This was a great move, and something that I've seen many other developers do (although often implicitly, not explicitly) and is a great way to recover from the stresses of open software development. Take a break from the noise. Be gentle with yourself.

Today, I follow a healthy number of people on social media and am relatively engaged, but I was definitely overly engaged for a long time. It's easy for that to happen.

Delegation

When you have thousands and thousands of people actively using your software, or even just your coworkers, it's easy to get burned out when you're the sole point-of-contact for the project. So, I've learned to delegate and collaborate more in new ways.

With Requests, I have two co-maintainers that handle incoming issues and take care of things like security releases. This greatly reduces the stress the project once put on me, and helped them to become very active members of the Python community.

My Balance Today

All of this was many many years ago, and I have a good balance today. I spend a significant amount of my free time with my hobbies like music production and photography (I even released an album this year). I published a book this year, but not without the amazing help of another wonderful human being who did the majority of the work involved.

It's really important to have hobbies other than writing code.

I don't code recreationally as much as I'd like to, but am starting to once again. I don't feel like I'm missing out on anything, but my evenings are not 100% filled with code anymore, as they once were.

Open source is all about collaboration. If you're finding yourself burned out, maybe you need to find new ways to collaborate with others in helping you do what you do best. You'll find that there are others who are in a position to help you do what you do best by doing what they do best too.

Generated from kennethreitz.org • 2025