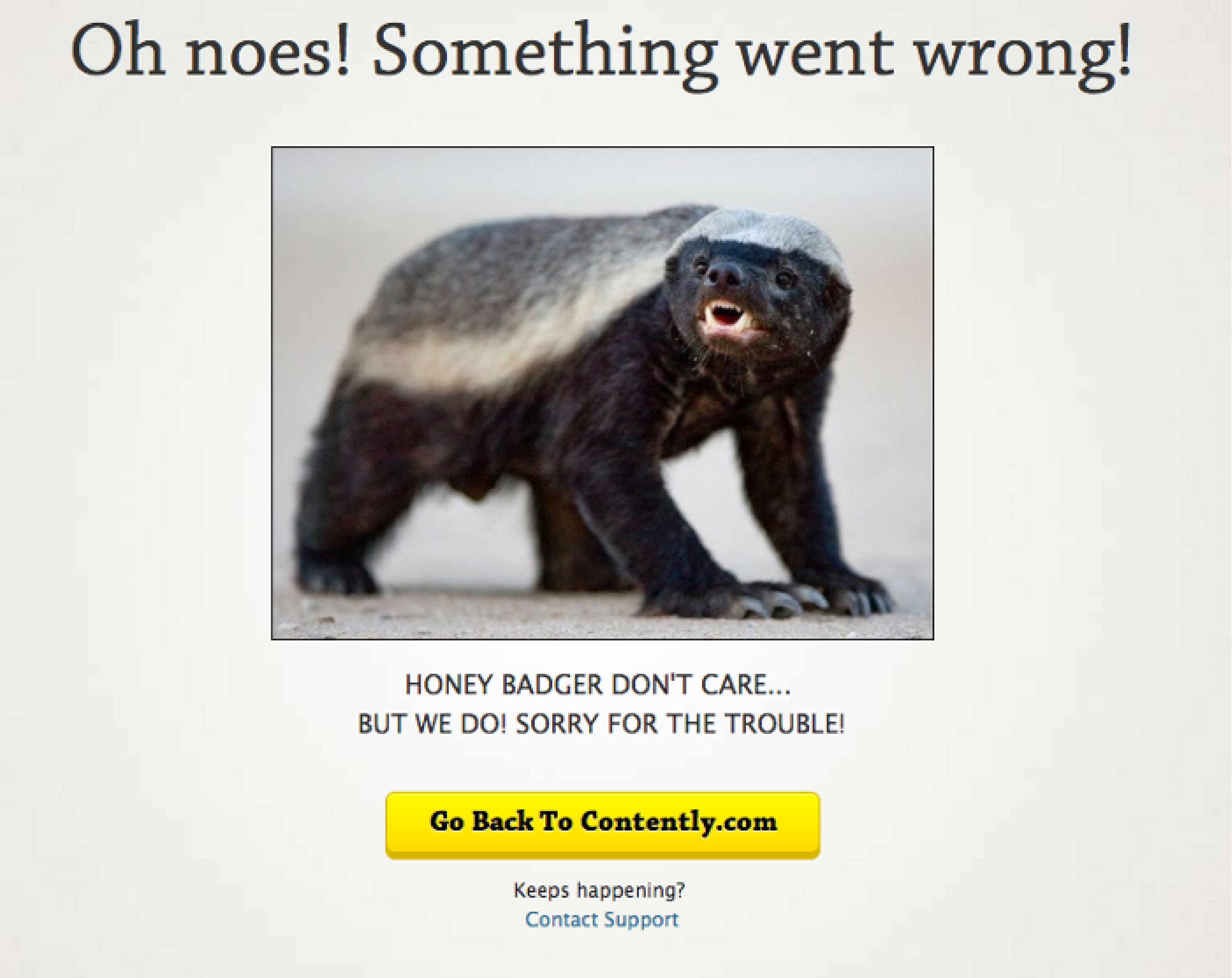
2 min read



Oh noes! Something went wrong!

HONEY BADGER DON'T CARE...
BUT WE DO! SORRY FOR THE TROUBLE!

Keeps happening?
Contact Support

Programming Is Failure

Sanjay Ginde in Tech Talk

Life is suffering.

Years ago I read the book Buddhism Without Beliefs and the above quote is essentially what I got from it. I was thinking about it in the context of programming earlier this week and realized I've internalized a similar mantra:

Programming is failure.

When I first learned to program I was so excited just to get something working, from that first "Hello World!" displayed in the terminal to a basic dynamic website.

But as getting things to work got easier, I found myself moving on to the next problem at the first sign of success. Inevitably, this led to sloppy code. Bugs would occur and, frustratingly, I'd go back and fix them. Or worse, someone else's design would fail and I′d have to rewrite and refactor chunks of my code. I was becoming an arrogant and resentful developer.

Over the years, though, I got more than used to those frustrations. I realized that ingrained in me was the above mantra. I′m now used to failure. I expect it to happen. No matter how much work or testing I do, it's just inevitable.

Now when a bug occurs, I (usually) look forward to fixing it. I′m curious to find the set of conditions I didn't foresee that caused a user's page to crash. It's no longer so much a chore to fix it, but a challenge.

And this mentality has become a part of the full software process for me. Now I expect a design to change. I no longer get as agitated when I complete a feature only to have to rework it because it doesn't quite work for the user. Not only do I strive to write more adaptable code, but I try to understand the interaction design and anticipate the problems with light prototypes.

It's affected the way I look at security as well. Sometimes my co-workers might think I'm overly paranoid, but I expect our site to get hacked. It's encouraged me to take on a security mindset and try to hack my own code. Can I manipulate some unique IDs in the URL or with Chrome's Web Inspector and gain access to other users' data? Can I change settings I'm not supposed to? What happens when somebody tries to break the site? I program with those expectations.

Perhaps the mantra that *programming is failure* may be overly pessimistic and discouraging for programmers. But I think it's realistic. I consider myself a pragmatic programmer and failure is just the nature of the work. The sooner a programmer accepts that, the sooner she will deal with all the issues that inevitably occur.

Suggest a link for further reading

Recommend