

Perfectionism

August 31, 2007

[Original link](#)

This may be hard for you to believe, but I'm something of a perfectionist. I get a special kind of joy out of tweaking every last detail until it's just right, making sure each pixel is in its place. Unfortunately, I've never really been able to indulge this desire on any of my previous projects, either because my skills are too weak (no matter how much time I spend on it, I'll never be able to craft a page like [John Gruber](#) can) or because time is too short.

At Reddit, for example, we were always rushing-rushing-rushing to get more done. In [a recent post](#), Paul Graham suggests startups can make you rich if you simply make sure to email him once a week. At Reddit we had no such worries: Paul was sure to email us weekly, sometimes even daily, usually asking why we hadn't added any new features to the site lately. (In fairness, it worked, I suppose.)

When you get that question once a week, you get to try out many different answers, including questioning the premise. At first I give the usual low-level excuses: we weren't feeling well this week, we had to go to a meeting, Alexis was out of town. Then we started in on more general principles: afraid of change, distracted by day-to-day fires, dissatisfaction with the codebase. And once I sent him [Slate's suggestion that](#) "the Eccentrics' own egotistical indolence has resulted in self-imposed limits to their skills—at the very least it deprives the world of more of their unique cultural prognostications." (He was not amused.)

Paul eventually became convinced that we had written lots of good code but wouldn't release it because we were perfectionists. *Knock it off*, he would tell us. *It's more important to get it up than to get it right*. Paul had become convinced that users love seeing new features, it gave them the impression of an exciting vibrant site.

There is something to this, of course. But I have a contrary proposal: users love perfectionism. Creating something brilliant is a process of continual refinement: adding bits where they're needed, chipping off others that aren't, and sanding everything smooth once it's in place, then polishing it until it gleams. Do one thing, release it, and don't stop releasing improvements until you do it really, really, well.

Adding features is part of this, of course, but not at all the whole thing. You can't create a great sculpture just by tossing on more and more clay.

Meanwhile, even if users can't immediately see all the subtle details you've added, they begin to add up to a feeling of — if you're talented — delight. The

software knows where you want to go and has already laid out the trail. The path is worn smooth and shiny. Everything looks beautiful along the way. And every day it gets a little bit better.

For my next startup, [Jottit](#), that's the plan.