

Schlep Blindness

paulgraham.com · Paul Graham · 2012-01 · [source](#)

| | |

|

January 2012

There are great startup ideas lying around unexploited right under our noses. One reason we don't see them is a phenomenon I call schlep blindness. Schlep was originally a Yiddish word but has passed into general use in the US. It means a tedious, unpleasant task.

No one likes schleps, but hackers especially dislike them. Most hackers who start startups wish they could do it by just writing some clever software, putting it on a server somewhere, and watching the money roll in—without ever having to talk to users, or negotiate with other companies, or deal with other people's broken code. Maybe that's possible, but I haven't seen it.

One of the many things we do at Y Combinator is teach hackers about the inevitability of schleps. No, you can't start a startup by just writing code. I remember going through this realization myself. There was a point in 1995 when I was still trying to convince myself I could start a company by just writing code. But I soon learned from experience that schleps are not merely inevitable, but pretty much what business consists of. A company is defined by the schleps it will undertake. And schleps should be dealt with the same way you'd deal with a cold swimming pool: just jump in. Which is not to say you should seek out unpleasant work per se, but that you should never shrink from it if it's on the path to something great.

The most dangerous thing about our dislike of schleps is that much of it is unconscious. Your unconscious won't even let you see ideas that involve painful schleps. That's schlep blindness.

The phenomenon isn't limited to startups. Most people don't consciously decide not to be in as good physical shape as Olympic athletes, for example. Their unconscious mind decides for them, shrinking from the work involved.

The most striking example I know of schlep blindness is Stripe, or rather Stripe's idea. For over a decade, every hacker who'd ever had to process payments online knew how painful the experience was. Thousands of people must have known about this problem. And yet when they started startups, they decided to build recipe sites, or aggregators for local events. Why? Why work on problems few care much about and no one will pay for, when you could fix one of the most important components of the world's infrastructure? Because schlep blindness prevented people from even considering the idea of fixing payments.

Probably no one who applied to Y Combinator to work on a recipe site began by asking "should we fix payments, or build a recipe site?" and chose the recipe site. Though the idea of fixing payments was right there in plain sight, they never saw it, because their unconscious mind shrank from the complications involved. You'd have to make deals with banks. How do you do that? Plus you're moving money, so you're going to have to deal with fraud, and people trying to break into your servers. Plus there are probably all sorts of regulations to comply with. It's a lot more intimidating to start a startup like this than a recipe site.

That scariness makes ambitious ideas doubly valuable. In addition to their intrinsic value, they're like undervalued stocks in the sense that there's less demand for them among founders. If you pick an ambitious idea, you'll have less competition, because everyone else will have been frightened off by the challenges involved. (This is also true of starting a startup generally.)

How do you overcome schlep blindness? Frankly, the most valuable antidote to schlep blindness is probably ignorance. Most successful founders would probably say that if they'd known when they were starting their company about the obstacles they'd have to overcome,

they might never have started it. Maybe that's one reason the most successful startups of all so often have young founders.

In practice the founders grow with the problems. But no one seems able to foresee that, not even older, more experienced founders.

So the reason younger founders have an advantage is that they make two mistakes that cancel each other out. They don't know how much they can grow, but they also don't know how much they'll need to. Older founders only make the first mistake.

Ignorance can't solve everything though. Some ideas so obviously entail alarming schleps that anyone can see them. How do you see ideas like that? The trick I recommend is to take yourself out of the picture. Instead of asking "what problem should I solve?" ask "what problem do I wish someone else would solve for me?" If someone who had to process payments before Stripe had tried asking that, Stripe would have been one of the first things they wished for.

It's too late now to be Stripe, but there's plenty still broken in the world, if you know how to see it.

Thanks to Sam Altman, Paul Buchheit, Patrick Collison, Aaron Iba, Jessica Livingston, Emmett Shear, and Harj Taggar for reading drafts of this.

|
|