The Other Half of "Artists Ship"

paulgraham.com · Paul Graham · 2008-11 · source

| | |

|
November 2008

One of the differences between big companies and startups is that big companies tend to have developed procedures to protect themselves against mistakes. A startup walks like a toddler, bashing into things and falling over all the time. A big company is more deliberate.

The gradual accumulation of checks in an organization is a kind of learning, based on disasters that have happened to it or others like it. After giving a contract to a supplier who goes bankrupt and fails to deliver, for example, a company might require all suppliers to prove they're solvent before submitting bids.

As companies grow they invariably get more such checks, either in response to disasters they've suffered, or (probably more often) by hiring people from bigger companies who bring with them customs for protecting against new types of disasters.

It's natural for organizations to learn from mistakes. The problem is, people who propose new checks almost never consider that the check itself has a cost.

Every check has a cost. For example, consider the case of making suppliers verify their solvency. Surely that's mere prudence? But in fact it could have substantial costs. There's obviously the direct cost in time of the people on both sides who supply and check proofs of the supplier's solvency. But the real costs are the ones you never hear about: the company that would be the best supplier, but doesn't bid because they can't spare the effort to get verified. Or the company that would be the best supplier, but falls just short

of the threshold for solvency—which will of course have been set on the high side, since there is no apparent cost of increasing it.

Whenever someone in an organization proposes to add a new check, they should have to explain not just the benefit but the cost. No matter how bad a job they did of analyzing it, this meta-check would at least remind everyone there had to be a cost, and send them looking for it.

If companies started doing that, they'd find some surprises. Joel Spolsky recently spoke at Y Combinator about selling software to corporate customers. He said that in most companies software costing up to about $1000 could be bought by individual managers without any additional approvals. Above that threshold, software purchases generally had to be approved by a committee. But babysitting this process was so expensive for software vendors that it didn't make sense to charge less than $50,000. Which means if you're making something you might otherwise have charged $5000 for, you have to sell it for $50,000 instead.

The purpose of the committee is presumably to ensure that the company doesn't waste money. And yet the result is that the company pays 10 times as much.

Checks on purchases will always be expensive, because the harder it is to sell something to you, the more it has to cost. And not merely linearly, either. If you're hard enough to sell to, the people who are best at making things don't want to bother. The only people who will sell to you are companies that specialize in selling to you. Then you've sunk to a whole new level of inefficiency. Market mechanisms no longer protect you, because the good suppliers are no longer in the market.

Such things happen constantly to the biggest organizations of all, governments. But checks instituted by governments can cause much worse problems than merely overpaying. Checks instituted by governments can cripple a country's whole economy. Up till about 1400, China was richer and more technologically advanced than Europe. One reason Europe pulled ahead was that the Chinese government restricted long trading voyages. So it was left to the Europeans

to explore and eventually to dominate the rest of the world, including China.

In more recent times, Sarbanes-Oxley has practically destroyed the US IPO market. That wasn't the intention of the legislators who wrote it. They just wanted to add a few more checks on public companies. But they forgot to consider the cost. They forgot that companies about to go public are usually rather stretched, and that the weight of a few extra checks that might be easy for General Electric to bear are enough to prevent younger companies from being public at all.

Once you start to think about the cost of checks, you can start to ask other interesting questions. Is the cost increasing or decreasing? Is it higher in some areas than others? Where does it increase discontinuously? If large organizations started to ask questions like that, they'd learn some frightening things.

I think the cost of checks may actually be increasing. The reason is that software plays an increasingly important role in companies, and the people who write software are particularly harmed by checks.

Programmers are unlike many types of workers in that the best ones actually prefer to work hard. This doesn't seem to be the case in most types of work. When I worked in fast food, we didn't prefer the busy times. And when I used to mow lawns, I definitely didn't prefer it when the grass was long after a week of rain.

Programmers, though, like it better when they write more code. Or more precisely, when they release more code. Programmers like to make a difference. Good ones, anyway.

For good programmers, one of the best things about working for a startup is that there are few checks on releases. In true startups, there are no external checks at all. If you have an idea for a new feature in the morning, you can write it and push it to the production servers before lunch. And when you can do that, you have more ideas.

At big companies, software has to go through various approvals

before it can be launched. And the cost of doing this can be enormous—in fact, discontinuous. I was talking recently to a group of three programmers whose startup had been acquired a few years before by a big company. When they'd been independent, they could release changes instantly. Now, they said, the absolute fastest they could get code released on the production servers was two weeks.

This didn't merely make them less productive. It made them hate working for the acquirer.

Here's a sign of how much programmers like to be able to work hard: these guys would have paid to be able to release code immediately, the way they used to. I asked them if they'd trade 10% of the acquisition price for the ability to release code immediately, and all three instantly said yes. Then I asked what was the maximum percentage of the acquisition price they'd trade for it. They said they didn't want to think about it, because they didn't want to know how high they'd go, but I got the impression it might be as much as half.

They'd have sacrificed hundreds of thousands of dollars, perhaps millions, just to be able to deliver more software to users. And you know what? It would have been perfectly safe to let them. In fact, the acquirer would have been better off; not only wouldn't these guys have broken anything, they'd have gotten a lot more done. So the acquirer is in fact getting worse performance at greater cost. Just like the committee approving software purchases.

And just as the greatest danger of being hard to sell to is not that you overpay but that the best suppliers won't even sell to you, the greatest danger of applying too many checks to your programmers is not that you'll make them unproductive, but that good programmers won't even want to work for you.

Steve Jobs's famous maxim "artists ship" works both ways. Artists aren't merely capable of shipping. They insist on it. So if you don't let people ship, you won't have any artists.