A Project of One's Own

paulgraham.com · Paul Graham · 2021-06 · source

| | |

|
June 2021

A few days ago, on the way home from school, my nine year old son told me he couldn't wait to get home to write more of the story he was working on. This made me as happy as anything I've heard him say — not just because he was excited about his story, but because he'd discovered this way of working. Working on a project of your own is as different from ordinary work as skating is from walking. It's more fun, but also much more productive.

What proportion of great work has been done by people who were skating in this sense? If not all of it, certainly a lot.

There is something special about working on a project of your own. I wouldn't say exactly that you're happier. A better word would be excited, or engaged. You're happy when things are going well, but often they aren't. When I'm writing an essay, most of the time I'm worried and puzzled: worried that the essay will turn out badly, and puzzled because I'm groping for some idea that I can't see clearly enough. Will I be able to pin it down with words? In the end I usually can, if I take long enough, but I'm never sure; the first few attempts often fail.

You have moments of happiness when things work out, but they don't last long, because then you're on to the next problem. So why do it at all? Because to the kind of people who like working this way, nothing else feels as right. You feel as if you're an animal in its natural habitat, doing what you were meant to do — not always happy, maybe, but awake and alive.

Many kids experience the excitement of working on projects of their

own. The hard part is making this converge with the work you do as an adult. And our customs make it harder. We treat "playing" and "hobbies" as qualitatively different from "work". It's not clear to a kid building a treehouse that there's a direct (though long) route from that to architecture or engineering. And instead of pointing out the route, we conceal it, by implicitly treating the stuff kids do as different from real work.
[1]

Instead of telling kids that their treehouses could be on the path to the work they do as adults, we tell them the path goes through school. And unfortunately schoolwork tends to be very different from working on projects of one's own. It's usually neither a project, nor one's own. So as school gets more serious, working on projects of one's own is something that survives, if at all, as a thin thread off to the side.

It's a bit sad to think of all the high school kids turning their backs on building treehouses and sitting in class dutifully learning about Darwin or Newton to pass some exam, when the work that made Darwin and Newton famous was actually closer in spirit to building treehouses than studying for exams.

If I had to choose between my kids getting good grades and working on ambitious projects of their own, I'd pick the projects. And not because I'm an indulgent parent, but because I've been on the other end and I know which has more predictive value. When I was picking startups for Y Combinator, I didn't care about applicants' grades. But if they'd worked on projects of their own, I wanted to hear all about those.
[2]

It may be inevitable that school is the way it is. I'm not saying we have to redesign it (though I'm not saying we don't), just that we should understand what it does to our attitudes to work — that it steers us toward the dutiful plodding kind of work, often using competition as bait, and away from skating.

There are occasionally times when schoolwork becomes a project of one's own. Whenever I had to write a paper, that would become a

project of my own — except in English classes, ironically, because the things one has to write in English classes are so bogus. And when I got to college and started taking CS classes, the programs I had to write became projects of my own. Whenever I was writing or programming, I was usually skating, and that has been true ever since.

So where exactly is the edge of projects of one's own? That's an interesting question, partly because the answer is so complicated, and partly because there's so much at stake. There turn out to be two senses in which work can be one's own: 1) that you're doing it voluntarily, rather than merely because someone told you to, and 2) that you're doing it by yourself.

The edge of the former is quite sharp. People who care a lot about their work are usually very sensitive to the difference between pulling, and being pushed, and work tends to fall into one category or the other. But the test isn't simply whether you're told to do something. You can choose to do something you're told to do. Indeed, you can own it far more thoroughly than the person who told you to do it.

For example, math homework is for most people something they're told to do. But for my father, who was a mathematician, it wasn't. Most of us think of the problems in a math book as a way to test or develop our knowledge of the material explained in each section. But to my father the problems were the part that mattered, and the text was merely a sort of annotation. Whenever he got a new math book it was to him like being given a puzzle: here was a new set of problems to solve, and he'd immediately set about solving all of them.

The other sense of a project being one's own — working on it by oneself — has a much softer edge. It shades gradually into collaboration. And interestingly, it shades into collaboration in two different ways. One way to collaborate is to share a single project. For example, when two mathematicians collaborate on a proof that takes shape in the course of a conversation between them. The other way is when multiple people work on separate projects of their

own that fit together like a jigsaw puzzle. For example, when one person writes the text of a book and another does the graphic design.
[3]

These two paths into collaboration can of course be combined. But under the right conditions, the excitement of working on a project of one's own can be preserved for quite a while before disintegrating into the turbulent flow of work in a large organization. Indeed, the history of successful organizations is partly the history of techniques for preserving that excitement.
[4]

The team that made the original Macintosh were a great example of this phenomenon. People like Burrell Smith and Andy Hertzfeld and Bill Atkinson and Susan Kare were not just following orders. They were not tennis balls hit by Steve Jobs, but rockets let loose by Steve Jobs. There was a lot of collaboration between them, but they all seem to have individually felt the excitement of working on a project of one's own.

In Andy Hertzfeld's book on the Macintosh, he describes how they'd come back into the office after dinner and work late into the night. People who've never experienced the thrill of working on a project they're excited about can't distinguish this kind of working long hours from the kind that happens in sweatshops and boiler rooms, but they're at opposite ends of the spectrum. That's why it's a mistake to insist dogmatically on "work/life balance." Indeed, the mere expression "work/life" embodies a mistake: it assumes work and life are distinct. For those to whom the word "work" automatically implies the dutiful plodding kind, they are. But for the skaters, the relationship between work and life would be better represented by a dash than a slash. I wouldn't want to work on anything that I didn't want to take over my life.

Of course, it's easier to achieve this level of motivation when you're making something like the Macintosh. It's easy for something new to feel like a project of your own. That's one of the reasons for the tendency programmers have to rewrite things that don't need rewriting, and to write their own versions of things that already exist. This sometimes alarms managers, and measured by total number

of characters typed, it's rarely the optimal solution. But it's not always driven simply by arrogance or cluelessness.
Writing code from scratch is also much more rewarding — so much more rewarding that a good programmer can end up net ahead, despite the shocking waste of characters. Indeed, it may be one of the advantages of capitalism that it encourages such rewriting. A company that needs software to do something can't use the software already written to do it at another company, and thus has to write their own, which often turns out better.
[5]

The natural alignment between skating and solving new problems is one of the reasons the payoffs from startups are so high. Not only is the market price of unsolved problems higher, you also get a discount on productivity when you work on them. In fact, you get a double increase in productivity: when you're doing a clean-sheet design, it's easier to recruit skaters, and they get to spend all their time skating.

Steve Jobs knew a thing or two about skaters from having watched Steve Wozniak. If you can find the right people, you only have to tell them what to do at the highest level. They'll handle the details. Indeed, they insist on it. For a project to feel like your own, you must have sufficient autonomy. You can't be working to order, or slowed down
by bureaucracy.

One way to ensure autonomy is not to have a boss at all. There are two ways to do that: to be the boss yourself, and to work on projects outside of work. Though they're at opposite ends of the scale financially, startups and open source projects have a lot in common, including the fact that they're often run by skaters. And indeed, there's a wormhole from one end of the scale to the other: one of the best ways to discover
startup ideas is to work on a project
just for fun.

If your projects are the kind that make money, it's easy to work on them. It's harder when they're not. And the hardest part, usually, is morale. That's where adults have it harder than kids. Kids just

plunge in and build their treehouse without worrying about whether they're wasting their time, or how it compares to other treehouses. And frankly we could learn a lot from kids here. The high standards most grownups have for "real" work do not always serve us well.

The most important phase in a project of one's own is at the beginning: when you go from thinking it might be cool to do x to actually doing x. And at that point high standards are not merely useless but positively harmful. There are a few people who start too many new projects, but far more, I suspect, who are deterred by fear of failure from starting projects that would have succeeded if they had.

But if we couldn't benefit as kids from the knowledge that our treehouses were on the path to grownup projects, we can at least benefit as grownups from knowing that our projects are on a path that stretches back to treehouses. Remember that careless confidence you had as a kid when starting something new? That would be a powerful thing to recapture.

If it's harder as adults to retain that kind of confidence, we at least tend to be more aware of what we're doing. Kids bounce, or are herded, from one kind of work to the next, barely realizing what's happening to them. Whereas we know more about different types of work and have more control over which we do. Ideally we can have the best of both worlds: to be deliberate in choosing to work on projects of our own, and carelessly confident in starting new ones.

Notes

[1]
"Hobby" is a curious word. Now it means work that isn't real

work — work that one is not to be judged by — but originally it just meant an obsession in a fairly general sense (even a political opinion, for example) that one metaphorically rode as a child rides a hobby-horse. It's hard to say if its recent, narrower meaning is a change for the better or the worse. For sure there are lots of false positives — lots of projects that end up being important but are dismissed initially as mere hobbies. But on the other hand, the concept provides valuable cover for projects in the early, ugly duckling phase.

[2]
Tiger parents, as parents so often do, are fighting the last war. Grades mattered more in the old days when the route to success was to acquire
credentials
while ascending some predefined ladder.
But it's just as well that their tactics are focused on grades. How awful it would be if they invaded the territory of projects, and thereby gave their kids a distaste for this kind of work by forcing them to do it. Grades are already a grim, fake world, and aren't harmed much by parental interference, but working on one's own projects is a more delicate, private thing that could be damaged very easily.

[3]
The complicated, gradual edge between working on one's own projects and collaborating with others is one reason there is so much disagreement about the idea of the "lone genius." In practice people collaborate (or not) in all kinds of different ways, but the idea of the lone genius is definitely not a myth. There's a core of truth to it that goes with a certain way of working.

[4]
Collaboration is powerful too. The optimal organization would combine collaboration and ownership in such a way as to do the least damage to each. Interestingly, companies and university departments approach this ideal from opposite directions: companies insist on collaboration, and occasionally also manage both to recruit skaters and allow them to skate, and university departments insist on the ability to do independent research (which is by custom treated as

skating, whether it is or not), and the people they hire collaborate as much as they choose.

[5]
If a company could design its software in such a way that the best newly arrived programmers always got a clean sheet, it could have a kind of eternal youth. That might not be impossible. If you had a software backbone defining a game with sufficiently clear rules, individual programmers could write their own players.

Thanks to Trevor Blackwell, Paul Buchheit, Andy Hertzfeld, Jessica Livingston, and Peter Norvig for reading drafts of this.

|
|