

## 4: Build Less

### Underdo your competition

Conventional wisdom says that to beat your competitors you need to one-up them. If they have four features, you need five (or 15, or 25). If they're spending x, you need to spend xx. If they have 20, you need 30.

This sort of one-upping Cold War mentality is a dead-end. It's an expensive, defensive, and paranoid way of building products. Defensive, paranoid companies can't think ahead, they can only think behind. They don't lead, they follow.

*If you want to build a company that follows, you might as well put down this book now.*

So what to do then? The answer is less. Do less than your competitors to beat them. Solve the simple problems and leave the hairy, difficult, nasty problems to everyone else. Instead of oneupping, try one-downing. Instead of outdoing, try underdoing.

We'll cover the concept of less throughout this book, but for starters, less means:

- Less features
- Less options/preferences
- Less people and corporate structure
- Less meetings and abstractions
- Less promises

# 5: What's Your Problem?

## Build software for yourself

A great way to build software is to start out by solving your own problems. You'll be the target audience and you'll know what's important and what's not. That gives you a great head start on delivering a breakout product.

The key here is understanding that you're not alone. If you're having this problem, it's likely hundreds of thousands of others are in the same boat. There's your market. Wasn't that easy?

Basecamp originated in a problem: As a design firm we needed a simple way to communicate with our clients about projects. We started out doing this via client extranets which we would update manually. But changing the html by hand every time a project needed to be updated just wasn't working. These project sites always seemed to go stale and eventually were abandoned. It was frustrating because it left us disorganized and left clients in the dark.

So we started looking at other options. Yet every tool we found either 1) didn't do what we needed or 2) was bloated with features we didn't need — like billing, strict access controls, charts, graphs, etc. We knew there had to be a better way so we decided to build our own.

When you solve your own problem, you create a tool that you're passionate about. And passion is key. Passion means you'll truly use it and care about it. And that's the best way to get others to feel passionate about it too.

## Scratching your own itch

*The Open Source world embraced this mantra a long time ago — they call it “scratching your own itch.” For the open source developers, it means they get the tools they want, delivered the way they want them. But the benefit goes much deeper.*

*As the designer or developer of a new application, you're faced with hundreds of micro-decisions each and every day: blue or green? One table or two? Static or dynamic? Abort or recover? How do we make these decisions? If it's something we recognize as being important, we might ask. The rest, we guess. And all that guessing builds up a kind of debt in our applications — an interconnected web of assumptions.*

*As a developer, I hate this. The knowledge of all these small-scale timebombs in the applications I write adds to my stress. Open Source developers, scratching their own itches, don't suffer this. Because they are their own users, they know the correct answers to 90% of the*

*decisions they have to make. I think this is one of the reasons folks come home after a hard day of coding and then work on open source: It's relaxing.*

—Dave Thomas, [The Pragmatic Programmers](#)

## **Born out of necessity**

*Campaign Monitor really was born out of necessity. For years we'd been frustrated by the quality of the email marketing options out there. One tool would do x and y but never z, the next had y and z nailed but just couldn't get x right. We couldn't win.*

*We decided to clear our schedule and have a go at building our dream email marketing tool. We consciously decided not to look at what everyone else was doing and instead build something that would make ours and our customer's lives a little easier.*

*As it turned out, we weren't the only ones who were unhappy with the options out there. We made a few modifications to the software so any design firm could use it and started spreading the word. In less than six months, thousands of designers were using Campaign Monitor to send email newsletters for themselves and their clients.*

—David Greiner, founder, [Campaign Monitor](#)

## **You need to care about it**

*When you write a book, you need to have more than an interesting story. You need to have a desire to tell the story. You need to be personally invested in some way. If you're going to live with something for two years, three years, the rest of your life, you need to care about it.*

—Malcolm Gladwell, author (from [A Few Thin Slices of Malcolm Gladwell](#))

# 6: Fund Yourself

## Outside money is plan B

The first priority of many startups is acquiring funding from investors. But remember, if you turn to outsiders for funding, you'll have to answer to them too. Expectations are raised. Investors want their money back — and quickly. The sad fact is cashing in often begins to trump building a quality product.

These days it doesn't take much to get rolling. Hardware is cheap and plenty of great infrastructure software is open source and free. And passion doesn't come with a price tag.

So do what you can with the cash on hand. Think hard and determine what's really essential and what you can do without. What can you do with three people instead of ten? What can you do with \$20k instead of \$100k? What can you do in three months instead of six? What can you do if you keep your day job and build your app on the side?

## Constraints force creativity

Run on limited resources and you'll be forced to reckon with constraints earlier and more intensely. And that's a good thing. Constraints drive innovation.

Constraints also force you to get your idea out in the wild sooner rather than later — another good thing. A month or two out of the gates you should have a pretty good idea of whether you're onto something or not. If you are, you'll be self-sustainable shortly and won't need external cash. If your idea's a lemon, it's time to go back to the drawing board. At least you know now as opposed to months (or years) down the road. And at least you can back out easily. Exit plans get a lot trickier once investors are involved.

If you're creating software just to make a quick buck, it will show. Truth is a quick payout is pretty unlikely. So focus on building a quality tool that you and your customers can live with for a long time.

## Two paths

*[Jake Walker started one company with investor money ([Disclive](#)) and one without ([The Show](#)). Here he discusses the differences between the two paths.]*

*The root of all the problems wasn't raising money itself, but everything that came along with it. The expectations are simply higher. People start taking salary, and the motivation is to build it*

*up and sell it, or find some other way for the initial investors to make their money back. In the case of the first company, we simply started acting much bigger than we were — out of necessity...*

*[With The Show] we realized that we could deliver a much better product with less costs, only with more time. And we gambled with a bit of our own money that people would be willing to wait for quality over speed. But the company has stayed (and will likely continue to be) a small operation. And ever since that first project, we've been fully self funded. With just a bit of creative terms from our vendors, we've never really need to put much of our own money into the operation at all. And the expectation isn't to grow and sell, but to grow for the sake of growth and to continue to benefit from it financially.*

—A comment from Signal vs. Noise

# 7: Fix Time and Budget, Flex Scope

## Launch on time and on budget

Here's an easy way to launch on time and on budget: keep them fixed. Never throw more time or money at a problem, just scale back the scope.

There's a myth that goes like this: we can launch on time, on budget, and on scope. It almost never happens and when it does quality often suffers.

If you can't fit everything in within the time and budget allotted then don't expand the time and budget. Instead, pull back the scope. There's always time to add stuff later — later is eternal, now is fleeting.

Launching something great that's a little smaller in scope than planned is better than launching something mediocre and full of holes because you had to hit some magical time, budget, and scope window. Leave the magic to Houdini. You've got a real business to run and a real product to deliver.

Here are the benefits of fixing time and budget, and keeping scope flexible:

### Prioritization

You have to figure out what's really important. What's going to make it into this initial release? This forces a constraint on you which will push you to make tough decisions instead of hemming and hawing.

### Reality

Setting expectations is key. If you try to fix time, budget, and scope, you won't be able to deliver at a high level of quality. Sure, you can probably deliver something, but is "something" what you really want to deliver?

### Flexibility

The ability to change is key. Having everything fixed makes it tough to change. Injecting scope flexibility will introduce options based on your real experience building the product. Flexibility is your friend.

Our recommendation: Scope down. It's better to make half a product than a half-assed product (more on this later).

## **One, two, three...**

| *How does a project get to be a year behind schedule? One day at a time.*

—Fred Brooks, software engineer and computer scientist

# 8: Have an Enemy

## Pick a fight

Sometimes the best way to know what your app should be is to know what it shouldn't be. Figure out your app's enemy and you'll shine a light on where you need to go.

When we decided to create project management software, we knew Microsoft Project was the gorilla in the room. Instead of fearing the gorilla, we used it as a motivator. We decided Basecamp would be something completely different, the anti-Project.

We realized project management isn't about charts, graphs, reports and statistics — it's about communication. It also isn't about a project manager sitting up high and broadcasting a project plan. It's about everyone taking responsibility together to make the project work.

Our enemy was the Project Management Dictators and the tools they used to crack the whip. We wanted to democratize project management — make it something everyone was a part of (including the client). Projects turn out better when everyone takes collective ownership of the process.

When it came to Writeboard, we knew there were competitors out there with lots of whizbang features. So we decided to emphasize a “no fuss” angle instead. We created an app that let people share and collaborate on ideas simply, without bogging them down with non-essential features. If it wasn't essential, we left it out. And in just three months after launch, over 100,000 Writeboards have been created.

When we started on Backpack our enemy was structure and rigid rules. People should be able to organize their information their own way — not based on a series of preformatted screens or a plethora of required form fields.

One bonus you get from having an enemy is a very clear marketing message. People are stoked by conflict. And they also understand a product by comparing it to others. With a chosen enemy, you're feeding people a story they want to hear. Not only will they understand your product better and faster, they'll take sides. And that's a sure-fire way to get attention and ignite passion.

Now with all that said, it's also important to not get too obsessed with the competition. Overanalyze other products and you'll start to limit the way you think. Take a look and then move on to your own vision and your own ideas.

## Don't follow the leader



*Marketers (and all human beings) are well trained to follow the leader. The natural instinct is to figure out what's working for the competition and then try to outdo it — to be cheaper than your competitor who competes on price, or faster than the competitor who competes on speed. The problem is that once a consumer has bought someone else's story and believes that lie, persuading the consumer to switch is the same as persuading him to admit he was wrong. And people hate admitting that they're wrong.*

*Instead, you must tell a different story and persuade listeners that your story is more important than the story they currently believe. If your competition is faster, you must be cheaper. If they sell the story of health, you must sell the story of convenience. Not just the positioning x/y axis sort of "We are cheaper" claim, but a real story that is completely different from the story that's already being told.*

—Seth Godin, author/entrepreneur (from Be a Better Liar)

## **What's the key problem?**

*One of the quickest ways to get yourself into trouble is to look at what your competitors are doing. This has been especially true for us at BlinkList. Since we launched there have been about 10 other social bookmarking services that have been launched. Some people have even started to generate spreadsheets online with a detailed feature by feature comparison.*

*However, this can quickly lead one astray. Instead, we stay focused on the big picture and keep asking ourselves, what is the key problem we are trying to solve and how can we solve it.*

—Michael Reining, co-founder, MindValley & Blinklist

## 9: It Shouldn't be a Chore

### Your passion — or lack of — will shine through

The less your app is a chore to build, the better it will be. Keep it small and manageable so you can actually enjoy the process.

If your app doesn't excite you, something's wrong. If you're only working on it in order to cash out, it will show. Likewise, if you feel passionately about your app, it will come through in the final product. People can read between the lines.

#### The presence of passion

*In design, where meaning is often controversially subjective or painfully inscrutable, few things are more apparent and lucid than the presence of passion. This is true whether the design of a product delights you or leaves you cold; in either case it's difficult not to detect the emotional investment of the hands that built it.*

*Enthusiasm manifests itself readily of course, but indifference is equally indelible. If your commitment doesn't encompass a genuine passion for the work at hand, it becomes a void that is almost impossible to conceal, no matter how elaborately or attractively designed it is.*

—Khoi Vinh, [Subtraction.com](http://Subtraction.com)

#### The bakery

*American business at this point is really about developing an idea, making it profitable, selling it while it's profitable and then getting out or diversifying. It's just about sucking everything up. My idea was: Enjoy baking, sell your bread, people like it, sell more. Keep the bakery going because you're making good food and people are happy.*

—Ian MacKaye, member of Fugazi and co-owner of Dischord Records  
(from [Salon.com](http://Salon.com) People | [Ian MacKaye](http://IanMacKaye.com))