

# Software that goes against the grain - It doesn't have to be like this

Written by Seth Corker on Benevolent Bytes

Photo by Will Myers on Unsplash

Software as a service (SaaS) is the prevalent business model and delivery method for many startups. It's a revenue model that investors love and is considered necessary for startups to scale up and win big, but the tactics that startups use to create software this way, has tradeoffs. There's another way to build software that goes against the grain and focuses on staying small.

## Why there's another way

Real-world analogues exist. When I go to a café, I make various decisions: - Where is the closest café? - Which one am I most familiar with? - Which one has seating? - Do I want food with it or just coffee? - How much do I like the baristas there? - What coffee do I feel like today? You get the point. All these questions I ask myself in a few short seconds before I decide which café I go to. My decision is shaped by my environment and context. Think about the environment in which the café exists. There isn't just one café, or even a handful. There's varying degrees of size and speciality, from a local café run by a couple of people to a global organisation like Starbucks employing thousands. They *are* competitors, but they still coexist. Multiple different approaches can thrive, and I have choice, it's not a winner-takes-all situation. Should software be any different?

An analogy always breaks down at some point. I understand software isn't necessarily something you want to make decisions about every day, but I think there are countless things we could relearn from other areas of our life. Here are some misconceptions we can challenge to get to compete in the space.

## Software is only chosen once

Individuals and businesses often choose software for the long haul. Once you've found software that does what you need it to do and compare it with what's out there, you're probably going to stick with it for a while. Software doesn't seem like the plethora of cafés at all. It seems more akin to your favourite café, the one you go to every time without fail.

It doesn't have to be like this.

A café doesn't have the same level of engagement as software. They might have a stamp card or rewards system to encourage you to come back, but it's not typically the deciding factor. Software, however, is often built to keep its hooks in you: 1. You provide *lots* of personal information. 2. The most popular business model is software as a service (SaaS) which charges a monthly subscription without you having to lift a finger. 3. You invest your time and effort. I doubt you're providing your screeds of data and personal information to your barista or paying a subscription whether you go to the café or not. This allows you to go to a different café if you like, and why shouldn't software be like this too?

Here's an alternate approach to building software which aims to contradict the previous three hooks: 1. Only use personal information when necessary. 2. Choose alternate pricing models that make sense for the product. 3. Follow common patterns to create familiarity with customers and use common integrations and standards. These are goals that could be picked up by smaller companies to provide software that does something a bit different. Software that follows these ideals is more interchangeable and there's less burden switching between them.

## Data belongs in the software I choose

Software keeps its hooks in you in another way. The data *you* provide it. Without stepping too far into "old man yells at clouds" territory, interactions used to be primarily file-based. You open a program/app, and it saves a file to your computer. That file is your data, that you have to do what you like with. Move it between devices, manipulate it with other software etc. With the advent of SaaS and mobile operating systems like iOS and Android, files aren't as visible to users. Data lives inside the apps that manipulate it. You don't see your tweets in a directory on your phone, they just exist in the ether. They are locked in the cloud.

It doesn't have to be like this.

When I go to a café, it's primarily for a good flat white and maybe a bite to eat to accompany it. Cafés sell a commodity. I could go to a number of different cafés, have different experiences that all cost different prices. Modern software couples the software itself (the café) with your data (the coffee). Once you've chosen a piece of software, your data is stuck. This is what makes it hardest to switch between SaaS products.

The software industry is pretty good at creating standards. The problem is that we don't always use them. One of the greatest challenges when switching software is data portability. It's often not available or, if it is, not in a format that's immediately usable. Software that uses open formats, encourages data exchange and APIs allows for more flexible workflows as well as switching between software.

### **Only large companies at large scale can offer good software**

Large companies are large for a reason, hopefully it's because they provide a product that's so good that it garners countless users and continuous improvement. Their products might have the biggest feature sets and most investment. This is how software companies scale, they offer a good product that grows its user base to the point where they actually start looking to do other things. This is where the issue lies, a company doesn't typically have a single product. They start with one and grow to many. Eventually, the product you rely on isn't getting as much attention as it once was, it gets sold and consumed by a bigger company and the very worst-case scenario is they decide it's not profitable to keep running.

It doesn't have to be like this.

Large chains like Starbucks don't necessarily make the best coffee. They have the best presence, you can be halfway across the world and probably be able to find a Starbucks in a major city. Small cafés likely don't have the reach, but they can serve more interesting coffee, atmosphere and underserved locations. These cafés can coexist in competitive harmony and if one goes out of business, there's likely an alternative waiting to pick up the slack.

Maybe startups don't have to scale to the same level. They don't need to put growth above everything. There's plenty of opportunity to stay small or medium-sized rather than bet it all and try to win big. Software companies that are sustainable and profitable should have the right to exist without being pushed upward into a scale up or down into closure.

### **The path forward**

Today, there are so many ways to build, package, distribute and sell software, but so often we choose the monthly subscriptions. Companies try to lock you in and often own your data, making it difficult to switch and even if you could go elsewhere, it's often an insurmountable task. Companies that make great software can typically lose their way as they put growth and expansion above all else.

All this is to say, it doesn't have to be like this.

I want to see more artesian software. SaaS that makes sense with pricing that doesn't double year-on-year. Apps that are small, sharp, and profitable without alienating users. I hope to see software that's unique and doesn't have to cater to everyone, it can serve a hundred of customers and still thrive. The opportunities are endless, but we're so focused on one path. I want to see the other side. I want software that goes against the grain.