InfoWorld DeepDive

29 ways to triumph as an independent developer

ART: STEPHEN SAUER/SHUTTERSTOCK COMPOSITE

Deep Dive

29 ways to triumph as an independent developer

Here's how to cut your corporate ties and transform your coding chops into a thriving business that can go the distance

BY STEVEN A. LOWE

For software developers, the main advantages of corporate employment -- job security, a clear path for career advancement, retirement benefits, health insurance -- are rapidly disappearing. Yet the demand for programming skills and services has never been greater. It might just be prime time to pack up that cubicle and go independent.

Being an independent software developer can be a dream life. But dreams include nightmares, so it's best to be prepared.

Being your own boss, setting your own hours, working only on projects that interest you -- all while earning a lot more money -- are possible when you strike out on your own as an independent software developer.

Possible. Not guaranteed.

Here are 29 best practices to get your business up and running.

Being an independent software developer can be a dream life. **But dreams include nightmares, so it's best to be prepared.**

Above all, understand and accept that “going independent” means “starting a business,” and a business is very different from a job.

1 Know why you’re doing this

Before you write that letter of resignation, think about why you want to become an independent software developer. Do you want to work from home? Do you want to travel? Do you want more money, fewer hours, or both? What are your main motivations for making this change? Write them down somewhere because chances are all of your goals will not be met immediately. When things get tough, that list will remind you of why you are doing this, and that can make the difference between giving up and success.

Above all, understand and accept that “going independent” means “starting a business,” and a business is very different from a job.

2 Plan before you quit

Going independent can give you greater say in the kind of work you do. And in the world of software, there are many possibilities; you don’t have to do exactly what you do at your current job. Consulting, development, products, training, and other services are all viable paths to independence. Make a list of everything you can do, everything you’d like to do, and everything you might be interested in learning how to do. Having that list will help prepare you to recognize opportunities when they arise.

Similarly, make a list of all the ways you can find and attract clients. When you make the leap, you’ll want to be in a position where you have more offers than you can accept.

3 Securing your first contract is key

The first contract can be the hardest, or the easiest. It is not uncommon for someone to resign as an employee then turn right around and contract with their former employer, doing exactly the same job as before, just with a different financial arrangement -- but it is not guaranteed. Another option is to contact a recruiting firm and have them find you a position that you like -- and make the terms of engagement corp-to-corp. This is how I started: I lost my "secure" job, and frankly, it was the best thing that ever happened to me. I called a friendly recruiter when I got home and had my first independent gig two days later -- at almost twice the pay.

4 Diversify your business skill set

Your job responsibilities will include everything, not just coding. Going it alone means going it alone. You'll have to do all of the work that your current team and organization do for you. This applies to both technical skills and "soft" skills -- either you learn how to do it yourself, or you hire someone to do it for you. If you hire others to help you, you'll have to learn how to manage people at the very least, but ideally how to lead them. Communication skills may mean more to your bottom line than sheer coding talent.

Focusing only on the paying work in front of you and letting everything else slide is a good way to code yourself out of a job.

5 Deliver exceptional service and quality

This may seem like an obvious platitude, but consider that the average office worker is productive only three to six hours per day, according to recent surveys. The rest of the time is sucked away in meetings, emails, social media, and various other nonproductive distractions.

When you are the business, you can eat Twinkies, drink Mountain Dew, and surf the Web all day if you want to. But chances are you only get paid when things get done -- so getting things done consistently, efficiently, and well will become paramount to the survival of your business. This makes Reddit a lot less interesting than a paycheck. Happy clients are key, and consistently delivering extraordinary results makes them happy.

6 Sweat the small stuff

Many tasks don't pay directly, but can lead to business failure (not to mention legal/tax trouble) if omitted. Accounting, cash flow, collections, contracts, liability insurance, and so on are easy to forget or postpone -- don't!

Other small things may suddenly matter, like being awake during business hours, answering your phone promptly, and having suitable clothes to wear to meetings with prospects and clients. The hip slacker image that works so well in your laid-back office job may not fly when you and only you represent the entire business. People will perceive you according to their inherent expectations, and you cannot control or change that -- but you can be aware of it, and prepare accordingly.

7 Always be marketing

Focusing only on the paying work in front of you and letting everything else slide is a good way to code yourself out of a job. Pay attention to additional opportunities with current clients, ask for referrals, continually prospect, and keep your pipeline full.

Marketing and sales are not evil; they are necessary. Even if all you have is a résumé on a job site, that is still marketing -- and the product is you. Even the best of clients can have sudden downturns; you do not want to be at the mercy of one client. To paraphrase Naomi Dunford of ittybiz.com, "If the vast majority of your income depends on one client, you don't have a business, you have a job."

8 Get organized

Since you are responsible for everything, you must be organized. I recommend David Allen's Getting Things Done system, but any system used consistently will do.

Whatever you choose, get things out of your head and into the system, and review it consistently. It is far too easy to think that you can remember everything you need to be doing, and perhaps you can, for a while, but the effort is draining and wasteful. An organizational system is not supposed to be a straightjacket or a dictator; it is a tool. Apply your organizational system consistently to all of your areas of responsibility -- not just technical ones -- to ensure that you are always on top of everything that needs to be done.

9 Going independent is not a promotion

There is a huge difference between going independent and getting promoted to management. If you get promoted to management, you become responsible primarily for the work of others, along with a load of administrative issues that you may not enjoy. If you go independent, you get managerial responsibilities and keep all your technical responsibilities as well.

Many good programmers have been ruined when turned into managers; many excellent technical people are simply not happy in management roles. If this is you, that is fine, just be aware of it and plan accordingly -- perhaps by hiring an assistant or designating one day a week as "admin day." As an independent developer, you are both management and worker, and you must do both well to succeed.

In "The 4-Hour Work Week," Tim Ferriss recommends several tactics to reduce the work you must do to a bare minimum, while maintaining the same income level.

10 Get out of your own way

Don't be your own barrier to business growth. At some point you will hit a limit of how much work you can do personally and still keep up with the business, maintain your health and family/social ties, and generally keep from burning out. You can raise your rates, but only up to a point. Learn how to delegate, sooner rather than later.

Many businesses that are otherwise doing well reach a tipping point where they fail because the person in charge cannot delegate, and that bottleneck strangles the business. If things are falling behind because everyone is waiting on you to do something, take this as a warning sign.

11 Know when to scale back, and when to double down

When you hit your limit, you can contract or expand. To do less, consider firing "bad" clients, raising your rates, and making better use of your time. To expand, consider hiring help, delegating or eliminating tasks, and diversifying your products and services.

In "The 4-Hour Work Week," Tim Ferriss recommends several tactics to reduce the work you must do to a bare minimum, while maintaining the same income level. Some of these tactics will work for software developers, such as outsourcing administrative tasks to trusted assistants (virtual or otherwise), while other tactics are appropriate only for product-based businesses. The point is not to blindly follow anyone's advice -- especially strangers on the Internet -- but to know what your options are and choose wisely.

12 Consider the product

Products have a distinct advantage over services: Products scale. You can only sell an hour of your time once, and it's gone, but you can sell a product over and over, even while you sleep.

Of course, products have their own overhead: First you have to write it, then you have to sell it. But once it is selling well, the only drain on your time is support, and counting your money. Consider building products (that people actually want) while supporting yourself by selling your services. If you can transform some of your services into products, so much the better.

13 Brand and network

People need to know about you. Whether you call it promotion or marketing or branding, the requirement is the same: Get the word out, with a consistent image and message about the kind of services and quality you provide. You can start a blog, join the local Chamber of Commerce, answer questions in LinkedIn groups, post instructive videos to YouTube, and so on.

Choose a few ways that you are comfortable with, and start. Be creative, use your imagination, find ways to stand out that represent you well, but above all make sure the channels you choose are populated with likely prospects. Try several approaches, track where your leads come from, and then concentrate on the avenues that prove fruitful.

14 Replace yourself

Eventually you may decide that you've had enough of being responsible for every aspect of your business; that's the time to replace yourself. You can replace yourself with one person, if it's the right person. More likely, your responsibilities will be divided among multiple people. This can be good, especially if the people you find share your philosophy and work ethic and are better at their jobs than you are.

Finding the right people can take a lot of effort, and you may have to go through a few to find the right one. It's best to test and fire early rather than invest too much time in trying to train or educate someone who is not up to your standards.

High-quality code is vital to independent developer success, but heads-down coding is a job, not a business.

15 Take care of your primary asset

The primary asset of your business is you. If you become burned out, or ill, or your skills get out of date, your business will suffer.

There are no sick days, there are rarely vacations, and deadlines don't care how you feel.

Be aware of new developments in your field, but spend R&D time wisely -- not only on things your clients want, but also on things that excite you.

Occasionally turn off the computer and go outside. Socialize with friends, spend quality time with family and loved ones, read a nontechnical book, and enjoy being human.

Most importantly, when the going gets rough, take time to recharge and remember why you started this journey. The payoff of time off to refocus is tremendous.

16 Don't code yourself out of business

High-quality code is vital to independent developer success, but heads-down coding is a job, not a business.

To be certain, you should always be thorough, with an eye toward delivering the best code you can. But make sure you have something else in the pipeline to work on when you're done.

When coding, it's all too easy to focus only on the work in front of you. In fact, it's far more efficient to devote yourself to coding than it is to task-switch between development, marketing, sales, planning, administration, and so on. But that's exactly what you will have to do to succeed. Billable work pays the bills, but finding more work keeps your business afloat.

17 Never forget that you are your brand

You may not realize it starting out, but even if you are working solo as an outside contractor with only one client, you have a brand, and that brand is you. Remember: A brand is a promise of quality and consistency, and though it can stand for whatever you think is important as a developer, it must be precise.

How you present yourself, your work, your team, and your development philosophy is the bedrock of your brand, and it should be reflected in all facets of your business, from your website to your business card to how you discuss your work with a casual acquaintance on a day off. If you are vague or inconsistent, your business will suffer.

Too often, independent software developers define their business by a tool or fad. Brands built around tools quickly become commodities -- would you rather hire a "Java shop" or an "expert in enterprise applications systems development"? Brands built around the latest trends can be lucrative, temporarily, but they too quickly become commodities.

The more specific you can be about your work and what you stand for, the easier it is to distinguish yourself from others, and the easier it is to identify and locate potential clients. More important, establishing a clear brand makes it easier for potential clients to identify and locate you.

18 Remember that you can't be everything to everybody

Customer satisfaction is essential to your success as an independent developer, but trying to please everybody can sink your business quickly. It causes you to spread your resources too thin, in the end pleasing no one.

This doesn't mean you shouldn't go the extra mile to serve clients to the best of your ability, or never take on work outside of your comfort zone. But pay attention to the limits of your abilities and resources. Overcommitting or signing up for work you can't complete to the level your customers expect can quickly poison your business.

Instead, be diligent about logging your work, so you can be accurate in estimating your bandwidth when special requests arise. Increment your experimentation into new areas rather than promise to deliver big, merely because you know you can learn new tools quickly. Remember -- you have a business to run, and mastering tech to the level of your brand takes time away from paying work.

Be diligent about logging your work, so you can be accurate in estimating your bandwidth when special requests arise.

19 Be wary of targeting the wrong market

Small businesses struggling to grow out of manual systems may seem like the ideal client for your fledgling independent software business; after all, they stand to reap the greatest benefits from automation. But they can also be the least trusting of technology and outsiders, and their cash flow may not be consistent enough to afford your services.

Targeting the "right" market is more art than science, but one vastly oversimplified maxim is to seek a combination of fit, finance, and fearlessness. Finding the best fit for your services is a complex, volatile process of ambitions, problems, needs, skills, gaps, benefits, and timing. There is likely to be no external signals for these, so you have to talk to a lot of people in various industries and organizations to find out. As you are doing this, you can assess the financial issue -- if an industry or organization can't afford your services, the whole process is pointless.

Gauging "fearlessness" is more of a gut instinct. Do the players in your targeted market have the organizational courage to see your partnership through? Software systems change business operations; for many industries, and many organizations, change is scary. Don't get lost chasing timid leads.

20 Develop the ability to identify any project's "champion"

Every software project has a "champion" (or "patron" if you prefer). This is the person with the authority to write the checks to pay for your work and the will to defend your project internally to ensure it succeeds. It is essential to identify this person and cultivate a meaningful relationship, where possible.

All too often the champion is not present, revealed, or even mentioned in consultative meetings. Worse, department heads or senior executives often insist they have the authority and budget to get the project done, when in fact that is not the case.

Always look at who signs the checks. In some companies, department managers have discretionary budgets and are indeed the champion of their projects. In other companies, executives higher up the chain -- sometimes more than one -- are the true, hidden champions; their approval is required for project expenditures.

It's not unusual, especially in smaller, privately held businesses, for the company president to be presented as the project champion, but in reality the CFO or company owner has to sign the checks. Don't be fooled into thinking you have a stronger commitment than you actually have.

21 Establish (and maintain) a sales pipeline

Selling custom software development is a slow, consultative process. Prospects may know they have a problem, but they may not be sure they really want to solve it. After all, the decision to adopt a custom solution can change the direction and foundation of an entire business; the bigger the business, the longer it takes to decide and turn. Even highly qualified, eager referrals from happy clients can take several months to close.

This means you must always add prospects to your sales pipeline. You must also always measure time spent and results. These numbers are crucial to keeping your pipeline flowing.

Suppose you need one client to support your business for half a year, and you can close one sale in six months for every 600 prospects you encounter. In this simplistic example, fewer than 100 prospects entering your pipeline every month could mean trouble.

In reality, the numbers are more complicated, but the structure of the pipeline remains the same: prospects in, time elapsed, sales out. Always be tracking time to qualify, time to close, average deal size, and close percentages. Use these numbers to project future income from the current pipeline. Subtract future costs and allow for projects (and income) ending. This will give you a leading indicator: If the resulting number is negative or trending downward, you are at risk of insufficient cash flow unless you add more clients to the pipeline.

22 Diversify your client base

Most independent software developers get their start with one main client. This can be a great way to strike out on your own, but beware attempting to maintain a business this way. When that one client has a hiccup, you'll be scrambling -- or out of business.

The other downside of having one main client is that they may start thinking of you as "staff augmentation" instead of "valued business partner," with demands going up and respect going down simultaneously. Three medium clients are better than one large one. Twenty small clients may be better still, as it is highly unlikely that a majority of them will suddenly stop at once.

Think of it this way: One client is a job; multiple clients, with diverse revenue streams, comprise a business.

Think of it this way: One client is a job; multiple clients, with diverse revenue streams, comprise a business.

23 Always account for taxes and overhead

Most ISVs are LLCs or S-Corps, both of which are "pass-through" companies

There are no shortcuts to sales that work reliably, and each technique and tool alters your process in ways both subtle and gross.

in the United States. This means you have to account for and pay estimated taxes as you go, and usually even more at the end of the year; nothing is deducted for you. Even if you pay yourself a salary, your finances can still get complicated and surprising at the end of the year. A good CPA can help, but it takes consistent financial discipline to stay current.

Nothing can sink your business like inattention to taxes.

24 Do not play fast and loose with collections and cash overflow

Most independent development shops run "lean," as in "very little cash buffer to fall back on." This can turn a minor payment issue into a layoff-inducing company crisis.

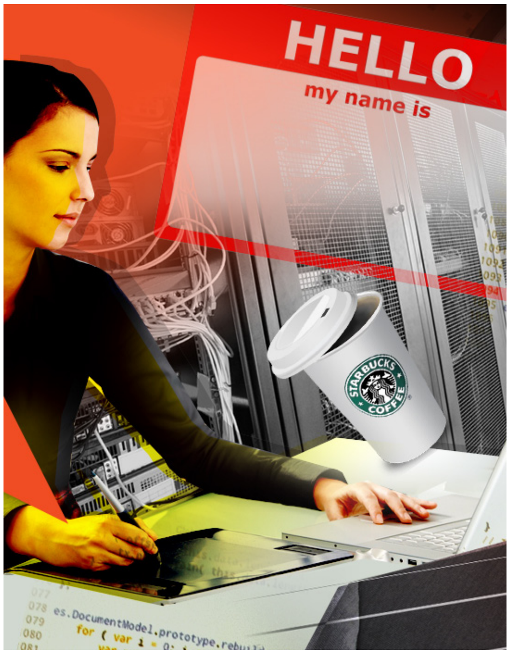
The fact is, clients don't always pay on time; this is normal, and it helps the relationship to be flexible. But if you notice a pattern -- unexplained delays, administrative misplacement of invoices, waiting on someone to sign checks, and so on -- the client may be having internal problems they aren't telling you about.

This may be a temporary issue, but it could last long enough to put you under water if you're not prepared for the possibility. Do your best to keep a buffer and pay close attention to collections. Understand the client's payment process, who is involved, and how long it takes; investigate immediately if the pattern starts to quaver.

25 Hire people only once you're sure you have enough paying work for them

A chief concern when going independent is knowing when it's time to no longer go it alone. While staying lean at the expense of adding help is the norm for some, it's nearly as easy to overestimate the scope or certainty of new projects and bring on eager, even inexpensive talent to help. After all, the only way to grow is to grow, right?

But this can increase your overhead significantly without increasing your revenue -- in terms of both budget and time. Changing processes to accommodate new head count, training new hires, managing workloads -- it can quickly derail your ability to close sales and deliver the quality your customers have come to expect. Hire cautiously.



26 Go talk to prospects – you really do have to learn the sales aspect of the business

Most developers who go independent do it to solve problems, write code, and deliver results. They may not enjoy marketing and sales activities, but these activities can't be ignored -- and I do not recommend that you outsource them completely, at least not initially.

There are no shortcuts to sales that work reliably, and each technique and tool alters your process in ways both subtle and gross. Long shots happen, but don't bet the farm on them; instead focus on steady, reliable, refinable systems that you thoroughly understand.

Evaluate each marketing/sales tactic and tool with the same critical eye you would use when deciding to adopt a new programming language or IDE. Play with it, run some tests and benchmarks, consider how it impacts everything else you're doing, and resist the urge to change directions too frequently. Even the most reliable, time-tested, surefire sales techniques take time and tweaking to work, so set realistic expectations and timelines, and be wary of abandoning existing systems before you understand them. If you hopped on the latest programming trend every month, you'd never finish anything and would quickly go out of business. The same applies to marketing/sales tools and trends.

Like it or not, there is no escaping the basic marketing/sales education process, learning curve, and effort. Go talk to prospects. There's no other way to learn this aspect of your business.

When you go it alone, every decision is important, and because it's your business, you're on the hook to make every one of them.

27 Document, refine and automate processes and systems

Processes and documentation may seem like the stuff of slow, dinosaur companies -- they may even be the chief reasons you left your corporate job -- but the truth is, an efficient independent software business depends on them.

Chances are you already have processes you have followed, repeated, and refined for years. Some you may have even taught, directly or indirectly, to others. Do yourself a favor: Write them down, discuss and refine them often, and automate wherever possible.

Automation is your bread and butter. You probably extol the virtues of automation to your clients all day long. But your own internal systems are manual or nonexistent. This makes results somewhat unpredictable; worse, it wastes your time -- and for an entrepreneur, time is the most critical nonrenewable resource.

Manual or hacked-together systems are fine to start with, but don't stay satisfied with them. Removing tedious, repetitive work reduces friction across all processes.

Look first at where most of your nonproductive time goes, and automate that. "Automation" in this sense includes all options, from fully automatic software systems that do everything for you while you sleep, to virtual assistants to personal assistants to outsourcing specialists and any combination of the above.

28 Pay attention to how you're going to provide customer support

Poor support is a major reason why clients change service providers. The less support needed, the better, but being known for excellent support is pure gold -- not just because it maintains customer relationships after sales and delivery, but because outstanding support turns your customers into evangelists.

This doesn't only mean fixing problems in the middle of the night. It means helping clients understand every facet of your solution, teaching them how to perform tasks and fixes themselves, and providing all of the information they need to remain confident they made the right choice for their business' future.

Because that's what you build for your clients: their future. You may not have to live in it for long, but they do. Support their decision and vision, and reinforce and reassure them. There is no such thing as "not my problem."

29 Learn how to delegate effectively

When you go it alone, every decision is important, and because it's your business, you're on the hook to make every one of them. As your company grows, the decisions mount -- which projects to pursue, how to juggle workloads, where to investigate new leads -- and they can quickly dam up and destroy the company.

With clear guidelines, responsibilities, and processes, delegation can be your most valuable tool, and one that can be comfortable even for the most control-freakish of development shop owners. ■

Steven A. Lowe *is a consultant, software developer, inventor, entrepreneur, author, musician, and lover of puns. He admits to being the author of From Burnout to Bonfire and a willing participant in the band Noise in the Basement, but neither confirms nor denies being the science-fiction author Steven Ayel.*