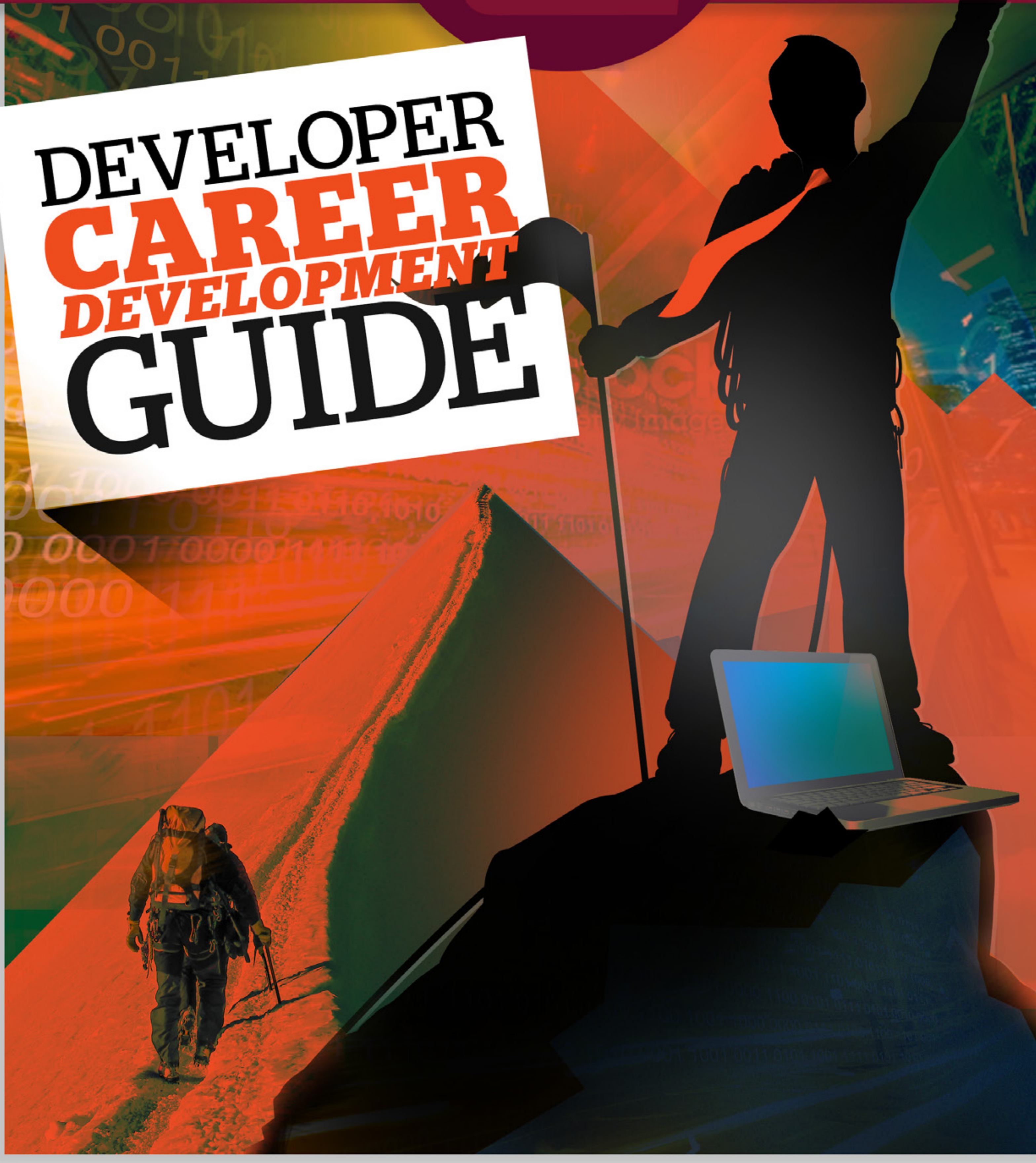
InfoWorld DeepDive
FROM IDG

DEVELOPER CAREER DEVELOPMENT GUIDE



THINKSTOCK

THE 13 DEVELOPER SKILLS YOU NEED TO MASTER NOW

From JavaScript to big data to devops, we break down your best bets for bolstering your career in the coming years. BY PAUL HELTZEL

Heavy weighs the crown of the developer king.

Yes, as software eats the world, demand for skilled developers remains high. But software's pervasiveness — from the server to the cloud to the coming onslaught of wearable and IoT devices — means far greater responsibilities for software engineers and the need to constantly expand your skills.

THINKSTOCK

Companies are now frequently looking for someone who is comfortable with every layer of the development stack, has the ability to extract insights from massive data sets, and can think strategically about devices to come, all while reconsidering old systems using the latest languages and frameworks. It's enough to have no idea where to start.

To find out the most sought-after developer skills this year, we reached out to a mix of recruiters, CTOs, CEOs, and other executives who offered their must-have technologies to try, strategies to consider, and soft skills to master.

If you're looking to dust off your resume or update your current skill set (you should be), let our breakdown of the most desirable skills and trending technology needs be your guide.

> **JavaScript has proven to be a highly portable and valuable skill set in today's market."**
>
> TODD ANGLIN, chief evangelist, Progress Software

Brush up on JavaScript

These days, developers who have mastered JavaScript can't go wrong, say those we surveyed. JavaScript proficiency is by far the most frequently sought skill named by executives and recruiters.

"Most developers already have an impression of the top keywords employers are searching for, having pored over job boards and comparative salary reports, while preparing for a job interview," says Sherif Abushadi, an instructor at Dev Bootcamp. "JavaScript is the talk of the town, as are dozens of related frameworks and libraries built by the JavaScript community."

Regardless of whether you're building for the desktop, the Web, or mobile, "JavaScript has proven to be a highly portable and valuable skill set in today's market," says Todd Anglin, chief evangelist at Progress Software.

Engineers with solid computer science fundamentals and knowledge of a modern stack won't go looking for work, says Mark Stagno, principal consultant for the software technology practice at WinterWyman Search. "This could be full-stack engineers or UI-focused developers who know JavaScript and a modern library such as AngularJS or React," he says.

Abushadi adds that, while JavaScript is king, other popular languages and approaches worth dipping your toes into these days include Ruby, in concert with the Ruby on Rails framework, and Python, in conjunction with Django; both technology stacks have proved themselves vital to building scalable Web applications.

Go big with data

Big data projects continued to get, well, bigger last year, and there's no sign of that slowing down in the years to come.

"While big data has been around for years, it's a trend that's here to stay," says Andrey Akselrod, co-founder and CTO of Smartling. "Developers must have in-depth knowledge of [business intelligence] and analytics products, machine learning tools, and other solutions that transfer, store, and aggregate large amounts of data. Only then can they help their organizations store, interact, and analyze big data to make better business decisions."

The pace of data creation is dizzying, says John Piekos, vice president of engineering at VoltDB. But so are the opportunities.

"Mobile and Internet of things devices are becoming ubiquitous worldwide," Piekos says. "Applications being developed today are harnessing amazing amounts of data and analyzing and reacting in real time. Technologies that capture and act on data the moment it arrives, such as streaming solutions and in-memory data stores, are becoming must-have skills to master. And technology that can store, manage, and historically analyze massive amounts of data — petabytes and up — will be skills that

> **Technologies that capture and act on data the moment it arrives, such as streaming solutions and in-memory data stores, are becoming must-have skills to master."**
>
> JOHN PIEKOS, vice president of engineering, VoltDB

serve developers well for the next decade."

For developers looking to add data wrangling to their arsenals, technologies such as Hadoop, Spark, R, and the variety of machine learning frameworks currently emerging are great places to start.

Master the full stack

Many top firms are now seeking full-stack developers who comfortably move between a variety of technologies and platforms.

These engineers "understand the implications of technical decisions from core layers of the software onto the presentation layer," says Hossein Rahnama, founder and chief product officer of startup Flybits. "These are great assets as they make the job much easier for their peers and will prevent the startup from developing silos by following a classical hierarchical technical decision-making. They enable the teams to remain small and effective. Leveraging platforms such as Top Coder and Amazon Mechanical Turks are also great ways to engage in cool projects."

Bryan Reinero, developer advocate at MongoDB, says going forward, engineers will need a broader range of skills to be effective: "Fortunately, increasing the scope of expertise is both healthy for the engineer as well as for the company in which she works."

Buy into devops

Some tech gurus think devops will fall by the wayside as the use of cloud computing continues to grow inside corporations. Not so, says MongoDB's Reinero.

"Devops skills are a clear stand-out," Reinero says, "often expressed by the dicta 'Engineers responsible for writing an application are the

10 steps to becoming the developer everyone wants

Even crack coder's need a higher gear to hit that next level

BY ANDREW C. OLIVER

You think it's all about programming, but you're wrong.

Great code is fine; commanding better work and a higher salary depends on ensuring more people know who you are. Plus, you may not think so now, but there will be times in your future when not every software developer who wants a job can get one. When the atmosphere becomes Darwinian, effective self-promoters do better than quiet toilers. You need to market yourself. Here's how.

TIP NO. 1: BLOG

Set up a blog, and post more than once a month. Do real research and make sure your writing is polished. Do the stuff your grade-school English teacher taught you: Create an outline, draw a narrative, check the grammar and spelling. Then, with great sadness, simplify and shorten it to the point where someone scanning it can make sense of what you're saying. The internet does not tolerate nuance.

TIP NO. 2: GO OPEN SOURCE

Don't believe the lies about open source. Even during the darkest stretches of the dot-bomb recession, all the developers on the open source project I started were quickly back at work. Just make sure the open source code you produce reflects the kind of job you want. I wanted to solve hard problems with the simplest solutions possible, but I've interviewed developers who, as was clear from their open source code, wanted to complicate simple problems. There's a market for that, but make sure your code reflects the market you're targeting.

THINKSTOCK

same engineers who maintain the application in production.' This includes the need to break down engineering silos such that engineers understand how their code operates in production and are mindful of performance and stability during the development phase."

> "Devops skills are a clear standout, often expressed by the dicta 'Engineers responsible for writing an application are the same engineers who maintain the application in production.'"
>
> BRYAN REINERO, developer advocate, MongoDB

Greater access to hot job prospects isn't the only reason to look into adding devops to your resume; devops practices simply make you a better developer and a more invaluable collaborator, Reinero argues.

"Engineers who think in these terms will release better code faster and with greater confidence," Reinero says. "Devops practices also improve team cohesion and operational agility. This is the kind of edge that allows a company to accelerate ahead of the pack."

The skills companies are looking for today are considerably more varied compared to a few years ago, says WinterWyman Search's Stagno: "Java and C# remain a part of the market, but when you look at companies founded after the last recession, you're seeing a variety: Ruby on Rails, Python/Django, Node.js, and the emergence of functional programming languages, with Scala the most prevalent.

"We're starting to see some companies

TIP NO. 3:

NOT SIX MONTHS, NOT 10 YEARS

Don't switch jobs every six months. When the job market cools, nothing will haunt you more than a job-hopping résumé. On the other hand, don't stay at the same place doing the same thing for 10 years. You'll become insulated and institutionalized. To stay valuable, be familiar with more than how to code IBM's stack while at IBM in the IBM way. Institutionalized devs often impress in interviews but fail the programming test.

TIP NO. 4:

EYE ON THE NEW STUFF, HANDS ON THE PRACTICAL

Young developers tend to work on the shiny. Ruby may be my favorite programming language, but it doesn't pay (on average) as much as Java, and the market is smaller. This may not always be true. Scala looks like it's coming on strong, but don't kid yourself about the market size — it isn't here yet. Of course, don't stay still so long that you are the future equivalent of a COBOL or PowerBuilder developer either.

TIP NO. 5:

WRITE YOUR OWN DOCUMENTATION

I can't tell you how many times I've worked on a project, only to be pulled into an executive meeting because I wrote a document or presentation they saw and understood. Always begin with an executive overview — the page you really have to read — and let the rest boil down the details. Ask yourself, What does a very busy person have to know about the topic if it's not the only thing they're working on? What most managers want to know: Who can drive this to completion and won't BS me about how it's going? Write that way, and it will pay off.

TIP NO. 6:

BREVITY IS THE SOUL

People who know what they're talking about tend to give shorter, more concise answers. So, in every communication, figure out how to summarize the issues. It's important to know the details, but don't load every sentence with minutiae, and by all means, don't build up hype — the sky probably isn't falling. When all else fails, lead with the money. Make sure your numbers are well thought out, plug them into charts, and clearly demonstrate that one point is superior to another in dollars and cents. If you can do that, you're destined to stand out among a lineup of pure coders.

adopt Go as well. I'm not a believer that you need to find the 'right' technology to master, but you want to make sure that you are current, as there are wrong technologies to choose that will put you behind the curve as the landscape is ever-changing."

Use the source

In particular for freelancers, the ability to point to your code on GitHub shows that your work has been put to good use and reviewed by your peers.

"Work on meaningful libraries and open-source them to demonstrate instantaneous value to potential employers," says Kiran Bondalapati, co-founder and CTO of ZeroStack, adding that contributing to open source projects can also help establish collaboration credentials.

Candace Murphy, recruiting manager at staffing services firm Addison Group, says that .Net and Java skills are still in great demand, but "larger trends in open source development are growing. We're seeing uptick in requests for IT professionals with Ruby, Python, Node.js, and AngularJS open source JavaScript experience. This trend is driven by companies moving away from the traditional platforms that require

"The skill of giving — and receiving — feedback is more often than not the key difference between successful and unsuccessful projects." SHERIF ABUSHADI, instructor, Dev Bootcamp

TIP NO. 7:

WOW THE CROWD

Figure out how to give presentations and learn how to speak in public. Research a topic and make yourself at least an expert, if not the expert. Presentations to the public are generally better if they are in part entertaining. It takes a lot of embarrassing mishaps to develop this skill, but an engineer who can explain the matter in plain English to management and give an expert talk on a topic will almost always command a higher salary than one who doesn't.

TIP NO. 8: BE REALISTIC

Maybe you like Erlang, but be realistic: The market for Erlang isn't big. You should know more than one language, as well as "new" or newly hyped topics, but you should avoid immature statements such as "I won't code unless it's in Erlang," unless you've truly considered the business issues. It can pay to be a narrowly focused expert, but even that has a cost — you'll be typecast according to your specialization, which may leave you high and dry when it's out of fashion. Sure, NoSQL is a better fit for your little project, but the company won't invest in it for a small one-off system, when the RDBMS will work just fine.

TIP NO. 9:

SOLVE THE HARD STUFF, KNOW THE TOOLS

Put in the time to learn a few tools other people don't commonly know. If you target the right ones, you will be more effective than other developers in your domain. For example, Aspect4j isn't for everyone, but it sure as heck is for me. I use it for things that are wrong — very wrong. I've rewritten .class file operations to make them run in Tomcat instead of WebSphere, though the original source was missing. I've fixed memory leaks in proprietary software. I've implemented a poor man's Wily Introscope. At each point, I looked like some kind of super-genius because I had a tool that few people had grokked yet.

TIP NO. 10:

PRACTICE HUMILITY

This is the least common skill among developers. Sometimes it means getting your hands dirtier than you want. Other times it means not letting it go to your head when you pack a room. Geek fame comes and goes, but remember, it's what you did recently that brings them in. Next week, it could all be gone. In the words of Tyler Durden, "You are not special." Yes, trolls, I'm fully aware of the irony.

licensing fees."

If companies themselves are exploring GitHub for technologies to add to their stacks, shouldn't you?

Be agile – and hone your teamwork skills

Agile development should be part of a coders' quiver of skills in 2016, says Greg Sterndale, co-founder of PromptWorks, a Philadelphia software consulting shop specializing in Web and mobile applications. And keep it simple: "Be humble and be hungry. Be familiar with agile and lean methodologies — the ability to break down big projects into small stories, prioritizing, adapting to change, and delivering the most value."

Since offering feedback is important in an agile environment, Dev Bootcamp instructor Abushadi stresses the need to understand your co-workers as well as the project: "The ability to provide honest, kind, and actionable feedback when working in teams is only truly possible when you have empathy, and the skill of giving — and receiving — feedback is more often than not the key difference between successful and unsuccessful projects."

Get secure

According to Addison's Murphy, companies that suffered security breaches last year already know what they want and what skills will be the most useful to them in 2016.

"They're taking more proactive approaches to increase security, not only within their IT department but across the board," Murphy says. "We'll see a shift in the most in-demand technologies this year as a result.

Experts noted the increased demand for network security, in particular adaptive application security, as well as cloud security.

"With the increase in adoption of the cloud, security and compliance are growing concerns for organizations," says Aashish Kalra, chairman of Cambridge Technology Enterprises. "This has resulted in an upsurge of demand for experts in security, compliance, governance, and data administration."

While developers may have traditionally passed the security buck to dedicated security pros, the need for developers to safeguard their code keeps growing. Consider it an invaluable, growing niche to fill.

Go mobile

Mobile developers are highly sought after, especially those who can distribute their creations widely, says Smartling's Akselrod. "I would argue that being a successful mobile developer is not achieved through a particular technology skill set," he says, "but rather through business savvy. Writing code is only the first phase of the project. Knowing how to promote your mobile app, and attract and retain customers, is what drives success."

"When mobile first emerged, the first-order problem was the app: How do I build these remarkable experiences that will run well across device types and operating systems?" says Jeff Haynie, CEO and co-founder of mobile technology company Appcelerator. "As the industry matured in this, acquiring better skills and tools, everyone now confronts the next big challenge: How do I get data — in the right format, of the right size, with the right resiliency and responsiveness — fed into these apps? That's a much bigger trick."

Connectivity issues and the fact that devices are moving beyond the screen further fuel the need to investigate the data piece of mobile development.

"The Web models of connectivity don't work," Haynie adds. "So this issue, coupled with

Everyone now confronts the next big challenge: How do I get data – in the right format, of the right size, with the right resiliency and responsiveness – fed into these apps? That's a much bigger trick."

JEFF HAYNIE, CEO and co-founder, Appcelerator the fact that more and more devices may not even have a screen and depend entirely on data and services for their usefulness, means that access and orchestration of data really is the new development challenge."

Mobile is one of the most attractive aspects of the Internet of things, says MongoDB's Reinero. But he warns it offers challenges.

"Mobile applications that start off small can become explosively popular and put heavy load demands on infrastructure," Reinero says. "This means that every back-end component of a mobile service, including application servers and database servers, must have a capacity to scale quickly. Engineers need to familiarize themselves with both with the scaling model of individual components ... as well as how to manage infrastructure-as-a-service offerings such as Amazon Web Services, Microsoft Azure, and Google Cloud Compute."

Even if mobile isn't your strong suit, familiarity with current and emerging technologies can boost your career.

"For mobile, developers with UX/UI experience are in high demand," says Jason Hayman, market research manager at TEKsystems. "The ability to understand and effectively collaborate with UX/UI pros can make developers stronger candidates."

I would argue that being a successful mobile developer is not achieved through a particular technology skill set, but rather through business savvy. Writing code is only the first phase of the project."

ANDREY AKSELROD, co-founder and CTO, Smartling

To the cloud

Unsurprisingly there's a steady demand for developers familiar with Amazon Web Services and Microsoft Azure. "In cloud providers, Amazon is still the biggest player by far, so keep up to date with their more advanced offerings like the API Gateway, Lambda, and the Container Service," says Nic Benders, chief architect at New Relic.

But it's not all about tools, when it comes to developing career opportunities in the cloud. In part of an ongoing trend, companies are looking for developers with business skills, including project management and the ability to negotiate with vendors, says TEKsystems' Hayman. "Additionally, there's a need for more 'move the business forward'-type skills, but less of a need for tactical work, as cloud providers are now increasingly responsible for that," he says.

"Success in the cloud means having deployed infrastructure that is secure, properly monitored, and properly managed," says MongoDB's Reinero. "IaaS and cloud platforms offer terrific opportunities, but improper management of a distributed cloud infrastructure can evaporate any advantage if failures exhaust a team's time and budget, and lead to unnecessary loss of business availability."

IoT: Making connections

The long-heralded concept of the Internet of things is now showing up both as a hiring demand and as a skill talented engineers want to explore themselves. And it's not only for

There's a need for more 'move the business forward' type skills, but less of a need for tactical work, as cloud providers are now increasingly responsible for that."

JASON HAYMAN, market research manager, TEKsystems embedded systems engineers anymore.

"You can do it even as a JavaScript developer," says Flybits' Hossein Rahnama. "The advent of protocols such as Wi-Fi Halo and wearable and IoT devices opening lightweight SDKs, will open many opportunities for developers to go beyond displays and build things for their surroundings and environments. We will also see many hardware/software co-designs due to the advent of these tools."

MongoDB's Reinero sees new opportunities where medical devices and the cloud converge: "This includes more wearable devices used for outpatient treatment and care, and smaller devices used in diagnostics," he says. "These devices will enable us to learn more about ourselves and vexing disorders. Data aggregation and analysis will be a critical part of how these devices are used. The availability of scalable and robust nonrelational databases used in conjunction with analytics systems will allow professionals to analyze medical data at a scale not previously possible."

Be persuasive

What about soft skills? Our experts frequently raised the idea that the ability to reach across divisions is a top demand for new hires.

"Client management skills are important, particularly the ability to push back tactfully but convincingly when there are alternatives that deliver more value," says PromptWorks' Sterndale. "Also being able to educate clients about the nature of software, guiding them toward practices that will serve them best in the long run."

"It's more important than ever … to demonstrate strong communication skills in a business setting," says Addison Group's Murphy. "With data security and privacy becoming exceedingly important in the boardroom, IT pros must concisely present their work in an easy-to-understand manner."

"Client management skills are important, particularly the ability to push back tactfully but convincingly when there are alternatives that deliver more value."

GREG STERNDALE, co-founder, PromptWorks

Be flexible

If you're the 10x, full-stack developer on your team, there's more demand than supply for your services. But if you're starting out or making a career change, the right attitude can make all the difference in getting — and staying — hired.

"On more than one occasion, I've gotten feedback from hiring managers referencing an engineer who is not necessarily the most talented engineer on the team, but is among the most valuable because of their attitude," says WinterWyman Search's Stagno. "Being a good team member, having a willingness to be part of the solution rather than part of the problem, lending a helping hand to team members, volunteering ideas and efforts to improve the product or the culture are valuable to any team." ■

"Being a good team member, having a willingness to be part of the solution rather than part of the problem, lending a helping hand to team members, volunteering ideas and efforts to improve the product or the culture are valuable to any team."

MARK STAGNO, principal consultant, WinterWyman Search

Failure may lead to success, but unthinking complacency is a certain dev career killer. BY PAUL HELTZEL

7 DEADLY CAREER MISTAKES DEVELOPERS MAKE

You'll find no shortage of career motivational phrases surrounding failure: Fail fast, failure builds character, the key to success is failure, mistakes make you grow, never be afraid to fail. But the idea of mistaking your way to the top of the software industry is probably unsound. Every developer will have their share of missteps in a career but why not learn from others' experience — and avoid the costliest errors?

THINKSTOCK

Deep Dive

I recommend making a list of experiences and skills that you'd like to acquire and use it as a map, updating it at least annually."

MICHAEL HENDERSON, CTO, Talent Inc.

That's what we did: We talked with a number of tech pros who helped us identify areas where mistakes are easily avoided. Not surprising, the key to a solid dev career involves symmetry: Not staying with one stack or job too long, for example, but then again not switching languages and employers so often that you raise red flags.

Here are some of the most notable career traps for engineers — a minefield you can easily avoid while you navigate a tech market that's constantly changing.

MISTAKE NO. 1: *STAYING TOO LONG*

These days it's rare to have a decades-long run as a developer at one firm. In many ways, it's a badge of honor, showing your importance to the business or at least your ability to survive and thrive. But those who have built a career at only one company may suddenly find themselves on the wrong end of downsizing or "rightsizing," depending on the buzzword favored at the time.

Opinions vary on how long you should stay in one place. Praveen Puri, a management consultant who spent 25 years as a developer and project manager before starting his own firm, isn't afraid to throw out some numbers.

"The longer you stay in one position, the more your skills and pay stagnate, and you will get bored and restless," Puri says. "On the other hand, if you switch multiple jobs after less than two years, it sends a red flag. In my own experience, I stayed too long on one job where I worked for 14 years — I should have left after six. I left other positions after an average of four years, which is probably about right."

Michael Henderson, CTO of Talent Inc., sees two major drawbacks of staying in one place too long. "First, you run the risk of limiting your exposure to new approaches and techniques," he says, "and secondly, your professional network won't be as deep or as varied as someone who changes teams or companies."

Focusing too much on one stack used by your current employer obviously is great for the firm but maybe not for you.

"It's a benefit to other employers looking for a very specialized skill set, and every business is different," says Mehul Amin, director of engineering at Advanced Systems Concepts. "But this can limit your growth and knowledge in other areas. Obviously staying a few months at each job isn't a great look for your résumé, but employee turnover is pretty high these days and employers expect younger workers like recent college graduates to move around a bit before staying long-term at a company."

MISTAKE NO. 2: *JOB JUMPING*

Let's look at the flip side: Are you moving around too much? If that's a concern, you might ask whether you're really getting what you need from your time at a firm.

Charles Edge, director of professional services at Apple device management company JAMF Software, says hiring managers may balk if they're looking to place someone for a long time: "Conversely, if an organization burns through developers annually, bringing on an employee who has been at one company for 10 years might represent a challenging cultural fit. I spend a lot of time developing my staff, so I want them with me for a long time. Switching jobs can provide exposure to a lot of different techniques and technologies, though."

Those who move on too quickly may not get to see the entire lifecycle of the project, warns Ben Donohue, VP of engineering at MediaMath.

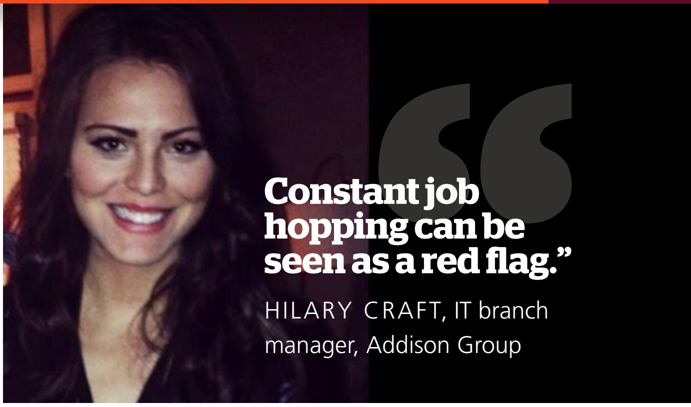
"The danger is becoming a mercenary, a hired gun, and you miss out on the opportunity to get a sense of ownership over a product

Soft skills and conversations with customers can also give a great sense of compassion that will improve how you build. You begin to think about what the customers really need instead of over-engineering."

BEN DONOHUE, VP of engineering, MediaMath

and build lasting relationships with people," Donohue says. "No matter how talented and knowledgeable you are as a technologist, you still need the ability to see things from the perspective of a user, and it takes time in a position to get to know user needs that your software addresses and how they are using your product."

Hilary Craft, IT branch manager at Addison Group, makes herself plain: "Constant job hopping can be seen as a red flag. Employers hire based on technical skill, dependability, and more often than not, culture fit. Stability and project completion often complement these hiring needs. For contractors, it's a good rule to complete each project before moving to the next role. Some professionals tend to 'rate shop' to earn the highest hourly rate possible, but in turn burn bridges, which won't pay off in the long run."



"Constant job hopping can be seen as a red flag."

HILARY CRAFT, IT branch manager, Addison Group

MISTAKE NO. 3: PASSING ON A PROMOTION

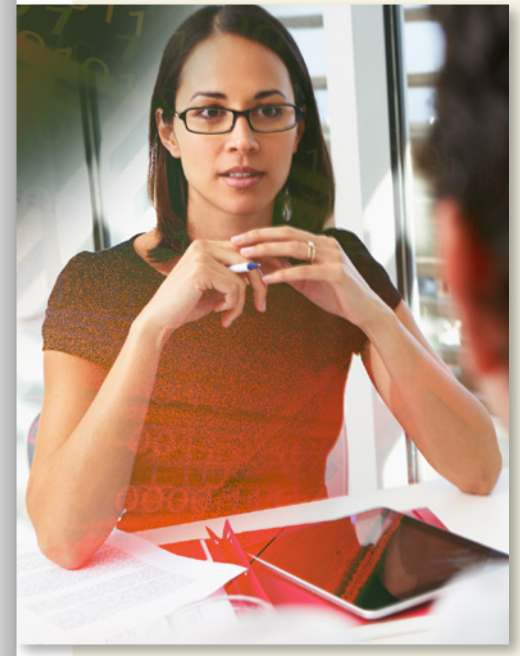
There's a point in every developer's life where you wonder: Is this it? If you enjoy coding more than running the show, you might wonder if staying put could stall your career.

"Moving into management should be a

13 tips for acing your interview

Esoteric puzzles, landmine questions, 'cultural fit' — these tips will help you navigate the interview process with confidence

BY STEVEN A. LOWE



THINKSTOCK

Job interviews can be grueling experiences, rife with esoteric puzzles, landmine questions, and the aching feeling that maybe you don't belong. You will talk with strangers via phone and video, take tests, answer challenging questions, travel on red-eyes, and endure long periods of not knowing what is going on, how you've been received, or what happens next.

To ace the interview and minimize your anxiety, you must prepare to an otherworldly extent. This means deep research on the position and company to decode what to expect during the interview process and to understand company dynamics at as near the level of a current employee as you can.

TIP NO. 1: DON'T BE AFRAID TO REACH OUT EARLY

Interview prep begins the moment you find yourself intrigued by a posting. The questions you anticipate fielding, how you'll present your skills and experience persuasively, what you'll ask to get a better grasp of the position — the interview should always be a point of reflection as you research the job. Even if the posting doesn't feel like an exact fit, don't be afraid to call or email. Early contact on a not-quite-right job can lead to an informal conversation about a better-suited job, or the possibility of the organization expanding the role to fit your qualifications.

cautious, thoughtful decision," says Talent Inc.'s Henderson. "Management is a career change — not the logical progression of the technical track — and requires a different set of skills. Also, I've seen many companies push good technical talent into management because the company thinks it's a reward for the employee, but it turns out to be a mistake for both the manager and the company."

Get to know your own work environment, says management consultant Puri, adding that there's no one-size-fits-all answer to this one.

"I've worked at some places where unhappy managers had no real power, were overloaded with paperwork and meetings, and had to play politics," Puri says. "In those environments, it would be better to stay in development. Long term, I would recommend that everyone gets into management, because development careers stall out after 20 years, and you will not receive much more compensation."

Another way of looking at this might be self-preservation. Scott Willson, product marketing director at Automic, asks the question: "Who will they put in your place? If not you, they may promote the most incompetent or obnoxious employee simply because losing their productivity from the trenches will not be as consequential as losing more qualified employees. Sometimes accepting a promotion can put you — and your colleagues/friends — in control of your workday happiness. Everyone should be in management at least once in their career if for nothing else than to gain insight into why and how manage-

"When senior developers don't have the time to mentor younger developers, I fully understand. Just don't say it's because 'I'm not good with people.'"

CHARLES EDGE, director of professional services, JAMF Software

TIP NO. 2:

DON'T BELIEVE EVRYTHING YOU READ ON GLASSDOOR

Resources such as Glassdoor provide a wealth of information about the hiring and interviewing process at most major companies, as well as reviews of the company's culture, salary information, and so on. Be aware that people who don't like the company are much more likely to review it than people who enjoyed their time there. Don't be dissuaded from pursuing what could be your dream job, or get derailed by interview advice from someone who might not have been the right fit or was not as well prepared as you will be.

TIP NO. 3:

FIND EMPLOYEE BLOGS AND READ THEM IN DEPTH

Most major companies have blogs, books, and websites devoted to their inner workings, technical focus, and business culture. Immerse yourself in these outlets to prepare for the interview, especially employee blogs, which can be a goldmine. Read them in depth, and soak in any YouTube presentations by employees as well. This give you a very good idea of the company's focus, technically and socially. The more you know about the company's inner workings, the easier it will be to convince interviewers that you belong and are ready to make a difference right away.

TIP NO. 4:

RESEARCH SOCIAL CULTURE – IT'S AS CRITICAL AS TECHNICAL FOCUS

Employee blogs, social media, and social networks are great sources of insider information about the company. Mine them to get an overall picture of the company's social culture as well to vet whether you you find it attractive and can see yourself fitting in. Many people take jobs they think they will like only to discover they don't enjoy the dynamics of the workplace once they are hired, or they enter the interview process blind to the social tenor the interviews will take. Getting a sense of the daily social interactions will give you a sense of what to expect from your interview.

ment and companies operate."

MISTAKE NO. 4: NOT PAYING IT FORWARD

A less obvious mistake might be staying too focused on your own career track without consideration of the junior developers in your office. Those who pair with young programmers are frequently tapped when a team needs leadership.

"I've found that mentoring junior developers has made me better at my job because you learn any subject deeper by teaching it than you do by any other method," says Automic's Willson. "Also, as developers often struggle with interpersonal skills, mentoring provides great opportunities to brush up on those people skills."

> **"Everyone should be in management at least once in their career if for nothing else than to gain insight into why and how management and companies operate."**
>
> SCOTT WILLSON,
> product marketing director, Automic

If experience is the best teacher, teaching others will only deepen your knowledge, says JAMF Software's Edge. That said, he doesn't hold it against a busy developer if it hasn't yet happened.

"Let's face it — no development team ever had enough resources to deliver what product management wants them to," Edge says. "When senior developers don't have the time to mentor younger developers, I fully understand. Just don't say it's because 'I'm not good with people.'"

TIP NO. 5: UNDERSTAND THE UNDERLYING PRINCIPLES OF INTERVIEW PUZZLES

One of the more controversial hiring practices these days is the use of puzzles, which themselves often find their way onto the internet. The last thing you should do is memorize the answers you find. Memorizing can easily backfire; just a small change to the question can render published answers incorrect. Plus, rattling off answers to complex questions without taking time to think is suspicious. Instead, try to assess the underlying purpose of the puzzles. Silly questions about cannibals and canoes may be intended to see how you think through a logic puzzle, or to see if you think out of the box, or to see if you push back against stupid questions. Each company has an agenda for posing puzzles; study published versions to unlock this agenda.

TIP NO. 6: CONNECT WITH CURRENT EMPLOYEES

As you prepare, make use of social networks like LinkedIn. Send an invitation to connect with a few people at the company along with a short note explaining that you have an interview and would appreciate some tips to confirm, deny, or expand on research you've done. You can ask about what to wear, what to expect, and so on, but limit correspondence to one question per contact; make it easy for them to connect and respond. Humor can help, but be yourself.

TIP NO. 7: DON'T TILT AT WINDMILLS

It takes a lot of work to prepare for an interview, and if your research reveals that a company's process or culture makes your skin crawl, walk away. Yes, practice makes perfect, and going through a not-quite-right or uncomfortable interview can be a learning experience. But chasing a bad fit can be unrewarding, demoralizing, and a distraction from finding your best fit.

MISTAKE NO. 5:

STICKING TO YOUR STACK

Your expertise in one stack may make you invaluable to your current workplace — but is it helping your career? Can it hurt to be too focused on only one stack?

MediaMath's Donohue doesn't pull any punches on this one: "Of course it is — there's no modern software engineering role in which you will use only one technology for the length of your career. If you take a Java developer that has been working in Java for 10 years, and all of a sudden they start working on a JavaScript application, they'll write it differently than someone with similar years of experience as a Python developer. Each technology that you learn influences your decisions. Some would argue that isn't a good thing — if you take a Java object-oriented approach to a loosely typed language like JavaScript, you'll try to make it do things that it isn't supposed to do."

It can hurt your trajectory to be too focused on one stack, says Talent Inc.'s Henderson, but maybe for different reasons than you think.

"Every stack will have a different culture and perspective, which ultimately will broaden and expedite your career growth," Henderson says. "For instance, I find that many C# developers are only aware of the Microsoft ecosystem, when there is a far larger world out there. Java has, arguably, the best ecosystem, and I often find that Java developers make the best C# developers because they have a wider perspective."

Automic's Willson says proficiency — but not mastery — with one stack should be the benchmark before moving onto another.

"It's time to move on when you are good at the skill, but not necessarily great," says Willson. "I'm not advocating mediocrity, just the opposite. I am saying that before you head off to learn a new skill make sure you are good, competent, or above average at that skill before

TIP NO. 8:

DRESS AS IF YOU ALREADY WORK THERE

How you present yourself influences interviewers, and thus the outcome of your interview. Dressing as if you already work at the company is important. If you're not sure, ask your recruiter or inside contact, look for office pictures, or drop by the office and peek through the window. Remind yourself of your good qualities, and enter the space as if you belong there — not as if you own it, but as if it is a comfortable place where your friends reside. Open, friendly confidence will set you and your interviewers at ease.

TIP NO. 9:

LET YOUR PERSONALITY OUT

Good interviewers look for someone who fits the company culture, adds value to the business, and is able to grow with the job. So let your personality out, though not all at once. Pick two or three areas of personal interest relevant to the job, company, or interviewer. You may not be able to identify these in advance, so stay alert for clues during the conversation. When opportunity presents, add a little bit of your personality to the dialog. If that increases interest, continue the thread for a bit; if it doesn't, don't press it.

TIP NO. 10:

BEWARE THE "INTERVIEWING TRAP" TRAP

Asking questions during your interview is important, but that old advice to take control of the interview by "interviewing the interviewer" can be misinterpreted as adversarial. Prepare some questions for the interviewer in advance, especially if you have concerns about the position or company, but by no means do you want to come off as challenging the interviewer by turning the process on its head. Aim to be remembered as well-prepared, well-informed, and easy to talk to — not eager to undermine colleagues for the sake of showing what you know.

you consider moving on."

Finally, Talent Inc.'s Henderson offers this warning: "Avoid the expectation trap that each new language is simply the old one with a different syntax. Developers of C# and Java who try to force JavaScript into a classical object-oriented approach have caused much pain."

MISTAKE NO. 6: NEGLECTING SOFT SKILLS

Programmers are typically less outgoing than, say, salespeople. No secret there. But soft skills can be picked up over time, and some of the nuances of developing a successful career — like learning from mentors and developing relationships — can be missing from your career until it's too late.

"It makes for better software when people talk," says MediaMath's Donohue. "Soft skills and conversations with customers can also give a great sense of compassion that will improve how you build. You begin to think about what the customers really need instead of overengineering."

Talent Inc.'s Henderson says your work with other people is a crucial part of developing a successful dev career.

"All human activities are social, and development is no exception," Henderson says. "I once witnessed an exchange on the Angular mailing list where a novice developer posted some code with questions. Within an hour — and through the help of five people — he had rock-solid idiomatic Angular code, a richer understanding of Angular nuance and pitfalls, and several new contacts. Although the trolls can sometimes cause us to lose faith, the world is full of amazing people who want to help one another."

Automic's Willson says a lack of soft skills is a career killer. Then when less proficient programmers move ahead developers who don't have people skills — or simply aren't exercising them

TIP NO. 11: HELP THE INTERVIEWER IMAGINE YOU IN THE POSITION

Open-ended questions are an excellent opportunity to help the interviewer imagine you in the position. For example, "What would my day-to-day duties be?" incites the interviewer to imagine you in the position going about your daily routine before answering, or at least drawing a parallel between you and their ideal candidate or the last person to fill the position.

TIP NO. 12: ALWAYS SPEAK FAVORABLY ABOUT FORMER EMPLOYERS

Even if your prior employer made your life hell, always speak favorably about prior employers. Going on about how badly a past employer treated you is a serious red flag, no matter how justified you may be. You will be asked point-blank why you left former positions, so be positive — "to use my abilities to help more people," "to take my career in a more modern direction." Whatever you say, mean it, and be ready to back it up with examples. Moving toward a positive goal is attractive; running away from a negative space is not.

TIP NO. 13: ASK FOR THE JOB

Most interviewers ask if you have questions at the end of the interview. If you are excited about the opportunity, it's OK to ask, "When do I start?" If the interviewer has hiring authority, he/she may make you an offer on the spot. This question shows enthusiasm and initiative, and at the very least elicits a laugh and ends the discussion on a positive note. If the interviewer reacts negatively, and it may be time to hunt anew, or to seek out ways to assuage their reservations about you. your code reflects the market you're targeting.

— are left wondering why. Yet everyone loves bosses, he says, "who demonstrate tact and proficient communication."

"To improve your soft skills, the Internet, e-courses, friends, and mentors are invaluable resources if ... you are humble and remain coachable," Willson says. "Besides, we will all reach a point in our career when we will need to lean on relationships for help. If no one is willing to stand in your corner, then you, not they, have a problem, and you need to address it. In my career, I have valued coachable people over uncoachable when I have had to make tough personnel decisions."

Programming is only one aspect of development, says management consultant Puri. "The big part is being able to communicate and understand business objectives and ideas, between groups of people with varying levels of technical skills. I've seen too many IT people who try to communicate too much technical detail when talking with management."

"The longer you stay in one position, the more your skills and pay stagnate, and you will get bored and restless."

PRAVEEN PURI, management consultant

MISTAKE NO. 7: FAILING TO DVELOP A CAREER ROAD MAP

Developing goals and returning to them over time — or conversely developing an agilelike, go-with-the-flow approach — both have their proponents.

"I engineer less for goals and more for systems that allow me to improve rapidly and seize opportunities as they arise," says Henderson. "That said, I recommend making a list of experiences and skills that you'd like to acquire and use it as a map, updating it at least annually. Knowing where you've been is as useful as knowing where you want to go."

And of course maybe equally as important — where you don't want to go.

"Early in my career, I hadn't learned to say no yet," says Edge, of JAMF Software. "So I agreed to a project plan that there was no way could be successfully delivered. And I knew it couldn't. If I had been more assertive, I could have influenced the plan that a bunch of nontechnical people made and saved my then-employer time and money, my co-workers a substantial amount of pain, and ultimately the relationship we had with the customer."

Automic's Willson gives a pep talk straight out of the playbook of University of Alabama's head football coach Nick Saban, who preaches having faith in your process: "The focus is in following a process of success and using that process as a benchmark to hold yourself accountable. To develop your process, you need to find mentors who have obtained what you wish to obtain. Learn what they did and why they did it, then personalize, tweak, and follow." ■

Deep Dive

THE PROGRAMMER'S GUIDE TO BREAKING INTO MANAGEMENT

The transition from command line to line-of-command requires a new mind-set — and a thick skin.

BY PAUL HELTZEL

Software development, like any career, is divided into leaders and producers. You're either Steve Jobs, or you're Woz. Two completely different approaches, and yet both can lead to great success. Talented engineers may see managing a team as the next step to growing their careers. So if you're moving in this direction, what tools do you need to make the transition? We'll look at some possible approaches, common pitfalls — and offer solutions.

THINKSTOCK

A first question might be whether to make a change at all. What if a Woz-like existence is more your style? Knowing yourself and whether management is really where you want to land is worth some self-reflection.

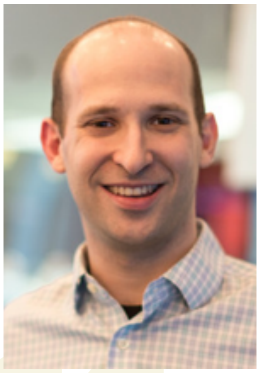
"You have to think about what aspects of the job you really enjoy, and which you try to avoid," says Adam Wolf, head of engineering for foundational applications at Bloomberg L.P. "If what you really enjoy doing is bringing everyone together to accomplish something as a team, or building a vision and getting everyone behind it, then management is a great opportunity to have a broader impact."

Consider the management transition thoroughly

The management track begins right where you are, in your current position. It requires taking on more responsibility, reaching out to team members, and making yourself visible. Because of this, you probably have a good first approximation of what to expect and an inkling of what will be expected of you, but there's a lot more to it than that.

Rick Hutley, clinical professor of analytics at University of the Pacific, advises checking how thick your skin is before you plan to manage others.

"Ask yourself how well you tolerate risk and criticism," says Hutley, a former CIO at British Telecom, and vice president of innovation at Cisco Systems. "Be honest. Better to be a happy grassroots worker than a miserable leader. That said, stretch yourself. Have the courage to move outside of your comfort zone and take on more responsibility."



"You have to be comfortable standing behind the work your team is doing."

ADAM WOLF, head of engineering for foundational applications, Bloomberg L.P.

Managing others will often lead to awkward situations. An exceptional career can be uncomfortable. And good managers are driven by a desire to lead and understand that delivering criticism may influence people, but maybe not win friends.

"Leadership means making hard decisions on occasions — disagreeing with those who used to be your colleagues — and it can be a lonely place," Hutley says. "The higher up you go the more certain it is you will fail — in someone's eyes."

James Casey, vice president of engineering at Seattle-based enterprise software firm Chef, says you can communicate your desire to move up the ladder — and this is critically important — by showing you have the qualities possessed by a good manager.

"Leadership and mentoring are at the top of the list — as an individual contributor," Casey says. "In parallel, ask yourself, 'Would my teammates want me leading this team? Why? Why not?' As you're thinking through how to move up, you need to be doing your current job very well — demonstrating that you can excel in your current responsibilities builds trust that you can move to a role with more responsibility."

But if you've never managed people before, how can you know if leading others is a good fit for you? Hutley offers these tips: "Are you one of those who tends to think beyond the immediate task, not just at work but socially as well? Do you suggest a better way of doing things or challenge things when they don't seem right? Do others seek you out for your thoughts or guidance? If this is you then you are a natural leader — and others recognize it too."

Communicate your plan

If you're hopeful hard work and attention to detail will speed your way to the top, you may need to broaden your plan. The leap to management will mean a complete redesign of your work life. But the first step is to telegraph your intentions to your boss.

Silicon Valley executive coach Jennifer Selby Long has spent two decades helping software developers move into management and effectively lead teams. Long says that great work isn't enough to send the right message — you'll need to be more up front. And it's not always easy.

"Nearly all software engineers want to remain software engineers, so your boss won't know that you want to get into management unless you say so directly," Long says. "Discuss your desire to contribute more to the company and take on a new challenge, not just in terms of how you want it for your own career benefit."

Performance reviews and other one-to-one meetings are a good time to lay out a desire to move up the ladder. And your manager may be able to help you determine what skills you need to develop to make it happen, says Chef engineering VP Casey.

"Regardless of whatever system your company uses for documenting your long-term career aspirations and directions, your one-on-one meetings with your manager are the best time to discuss your goals in detail," Casey says. "If you and your manager are both agreed on the path you need to take to reach management-level responsibility, then they will also be your biggest advocate. That's the best route to success."

And along with these other tips, if you're looking to make the move, one of the subtle yet painful changes might be changing the way you present yourself at the office. It's obviously a personal choice, and Mark Zuckerberg aside, if you're going to be running meetings your street clothes may need to go.

"The best advice I ever received was: 'If you want to be something — look like it,' says Hutley. "When a company wants to fill a position they look through their mental rolodex for someone who 'looks like' the person they are looking for. It's too late to start looking like a leader when the job advert comes out — you have already been pigeonholed by your actions up to that point."

And that subtle shift toward wearing the part — and separating yourself from the pack — may be enough for the friction to begin.

"Looking like a leader can itself be a little uncomfortable. Wearing neat pants and clean shirts when everyone around you is in jeans and sneakers can lead to some leg pulling," Hutley adds. "But you can't climb the ladder by standing on the bottom rung with everyone else. You have to differentiate yourself in the way you look, speak, and act."

"As a manager, your job will be 90 percent about influencing people, which is an inherently illogical task."

JENNIFER SELBY LONG, executive coach

Help along the way

You could, of course, apply to an MBA program and complete it online or after work. Public speaking courses can help, say our experts, along with budget training, self-assessments like Myers-Briggs, and training in diversity and inclusion. But there are plenty of opportunities at the office that can help you move in the right direction.

"I should disclaim this answer by saying that I don't have an MBA," says Bloomberg's Wolf. "My feeling is that I learned more about being a manager by actually being one, than by learning about the role. I have taken classes and read books on management, but I found the thing that helped me most was getting good feedback from my colleagues, managers, and my team, and from watching role models and trying to learn what made them effective."

"Find mentors," agrees Hutley. "These do not have to be people who have been formally assigned as mentors — although they are good too. Identify leaders you resonate with — who

"Look for the fastest-scaling areas of the business, since those areas will present a tremendous number of organic opportunities to step up and take on additional responsibilities."

SARAH NAHM, co-founder and CEO, Lever

display qualities you admire and wish to emulate. Then observe them whenever you can and understand why you admire them: How did they handle a particular situation; how do they dress, speak, act?"

Pursuing certification in your field can also show that you're looking to advance, says Eric Klein, managing director of staffing firm HireStrategy. And you can show leadership qualities by helping along new or junior colleagues.

"Suggest a peer code review when a colleague is stuck in development," Klein says. "Aside from managing projects and teams, mentoring junior staff and peers can demonstrate your ability to lead others."

Sarah Nahm, a former Chrome team member at Google, advises you to look for areas where your current business is growing.

"As a manager, your responsibility will be less about doing the work and more about helping others be successful."

MICHAEL DE GROOT, chief architect, Geneca

"Engineers should pay attention to the business around them," says Nahm, who now leads Lever, a Silicon Valley firm that helps companies hire well while scaling up. "Look for the fastest-scaling areas of the business, since those areas will present a tremendous number of organic opportunities to step up and take on additional responsibilities. Volunteer to do more interviewing and take a front-row seat to strategic hiring decisions. You'll see exactly what qualities are important to your engineering org, and also get a head-start on a critical skill for when you need to hire engineers for your future team."

Making the transition

One of our pros says management offers many of the same challenges and uncertainties as parenting. He then quickly adds you must never actually express that — or you'll risk alienating your entire team. Along with that handy analogy/warning, here's a blueprint for the transition.

First, get ready for "a complete and total career change," says executive coach Long. "There are no product specs or algorithms for people. As a manager, your job will be 90 percent about influencing people, which is an inherently illogical task, and dealing with ambiguity in the business while still producing results through others, which is also a task that can't be done by leaning on logic and reason alone."

And now for the really tough part. Are you ready to hand over control and let your team do their jobs?

"The worst managers are those who maintain too much control," says Long, "yet micromanagers always, always think they're doing the right thing. Can you dedicate yourself to guiding, supporting, directing, and advocating for others? At first, it's a white-knuckle ride. Be sure you're ready to get on that roller coaster."

And what if your team includes former colleagues from your workgroup? How do you retain a collegial relationship with these co-workers?

"You have to remain sufficiently detached to be objective and to make and communicate the hard decisions honestly," says Hutley. "It's very hard to give hard negative feedback honestly — and it's hard for both parties, not just the leader. That said, you don't have to completely shun your old colleagues."

Let your team stretch their skills, our pros say, and be ready to do some nail-biting when they fail. Again, these skills — communicating effectively, being persuasive, and keeping your cool during times of uncertainty — may not come naturally. But they can be learned.

"A leader guides and coaches and then lets the chips fall where they may," Hutley says. "Be ready to praise when things go right — something we forget to do all too often — and to support and encourage when things go wrong."

Bloomberg's Wolf agrees and advises new managers give their team the same autonomy they wanted before they made the jump.

"You can't go too far and abdicate responsibility," Wolf says. "You have to be comfortable standing behind the work your team is doing. I find it's helpful to focus on asking good questions of your team, and letting them reach the right answers."

And don't give up on what got you this far. The quickest way to become out of touch is to let your skills get rusty.

"Use mainstream technologies that are transferable from one company to the next and remain hands-on," says Mark Stagno, principal consultant at staffing firm WinterWyman. "In a 'what have you done for me lately' industry, your ticket ... is your technical skill-set. If you abandon that, you are vulnerable, and if things change — the company begins to struggle or you become unhappy — it won't be as easy to find a job if you aren't hands-on."

An argument for staying put

Let's also consider that engineers who move up may look back wistfully to a time when building and deploying code was the focus, rather than managing a product, budgets, and a team.

HireStrategy's Klein says he occasionally hears from engineers-turned-managers that they miss the hands-on work of coding. "With technology constantly changing and evolving, managers notice their technical skills slipping even as their management skills improve," Klein says. "We remind candidates that, at the end of the day, it's all about each individual finding what they love to do every day. Careers can excel without going down a management track."

Many engineers enjoy a solo approach to problem solving, says Michael de Groot, chief architect at software product development firm Geneca, and take pride in coming up with novel solutions on their own.

"As a manager, your responsibility will be less about doing the work and more about helping others be successful," de Groot says. "You'll have to deal with other people's behaviors, attitudes — and differences in work ethic."

And one final thought about moving up the ladder. In a time where rock-star developers are hard to find, being the boss may not mean better compensation. It's a far different world than when Woz looked around his calculator-producing colleagues at Hewlett-Packard — and thought he'd found a job for life. In today's market, those who can innovate are the ones in demand.

"Engineers at the top end of the market are now making the same if not more than a line manager," says WinterWyman's Stagno. "Think long and hard before taking the plunge into management, and above all else, make sure you are doing it for the right reason — that you want to be a leader rather than simply wanting to move up the career ladder." ■

"In a 'what have you done for me lately' industry, your ticket ... is your technical skill-set. If you abandon that, you are vulnerable."

MARK STAGNO, principal consultant, WinterWyman