

# Technical Hiring Guide



Learn to  
'speak geek'  
in 4 simple  
steps

Recruiting for technical roles can feel like being in a strange place and not speaking the language; the acronyms and jargon can make it challenging to navigate a candidate’s qualifications beyond matching buzzwords on their resume to a job description.

This Technical Hiring Guide explains different tech roles and how they relate to one another in plain English. Before we get into specifics, let’s take a top-down approach to the roles.

**STEP 1**

## Understand technical departments

First, it helps to understand the departments within a company, as different divisions have different goals and responsibilities. Most technical hires fall into one of two camps: Building Software or Operations and Support.

**Building Software** encompasses all the roles related to creating software. “Software” could mean desktop applications, mobile apps, websites, and other tools. Basically, the output or result from “writing code.”

**Operations and Support** represents the logistics needed to keep things running. “Things” could be software, hardware, or processes in a business or organization. A great example is IT or tech support teams – their primary job is to keep things running smoothly, rather than writing code or building software.

One big difference between these two categories is that Building Software often (though not always) generates products, features, or revenue, while Operations and Support is generally a cost center to a business. Both groups are essential, so it is important to know how their different goals fit into the overall organization.








**STEP 2**

## Know the team & tools

Once you understand the technical areas of the business, the next step is to gain a deeper understanding of the team and the open position. Get a solid handle on the technologies in use and the desired skillset of any new hire.

Since job descriptions don't always paint the whole picture, dig deeper into what the hiring manager is actually looking for in a new hire by asking key questions:

-  What is their technology stack?
-  What skills are their team missing?
-  What technologies or tools will this new hire use on the job?
  - Will the person be able to learn any of these on the job, or do they need to come with a mastery of one or several of those skills?
  - If a candidate doesn't have all the requisite skills, what other technologies or comparable skills might still make them qualified? If so, what are they?
-  What are the skillsets and backgrounds of the current team?
-  What are the skills, experiences, and traits of your most successful team members?



Not all hiring managers will be able to answer all of these questions, but this line of questioning will help you understand how eligible they consider different candidates, and help you identify the true “must-haves” in the position.

**STEP 3** Match resumes to roles

Now for the tricky part – matching candidates to roles.

One easy way to match a candidate with a job is to see if the candidate has a requisite skill on his or her resume. However, that doesn't necessarily guarantee a good fit. For example, a person may not have mobile programming

experience, but may have a strong C++ background, a deep desire to work in iOS, and a great attitude that could easily fit into the role of iOS engineer. To see this potential match, you have to understand the open role and all the candidate's previous positions.

In the appendix of this Guide, you will find some common technical job titles, overviews of what each job entails, along with sample interview questions and answers for each role. These will help you dive deeper into understanding what each position requires and better screen and filter candidates for the hiring manager.

Software Engineer .....	4
Back-End Engineer .....	4
Middle-Tier Engineer .....	5
Front-End Engineer .....	5
Web Developer .....	6
Database Administrator (DBA) .....	6
DevOps Engineer .....	7
System Administrator .....	7
Network Administrator .....	8
Data Scientist .....	9
Quality Assurance Engineer .....	9
Software Engineer in Test .....	10
Technical Lead .....	11

## STEP 4

### Ask good questions

Regardless of the resume or role you are hiring for, remember that every candidate is just a person with a collection of skills and strengths. Therefore, the best thing you can do is ask lots of questions and try to hone in on what makes each person special. Consider some of the following questions when you're meeting with a candidate:

- ▶ Could you put them in front of customers?
- ▶ Do they explain things well enough to talk to a CEO?
- ▶ Are they curious and have a strong desire to understand how things work?
- ▶ Have they ever been on-call before and would you rely on them?

For each role, dig into the must-haves versus what can be learned. Then devise specific questions to help draw that out from a resume. Try to create questions that specifically target the key aspects of the role the candidate is applying for. Rather than asking everyone if they are detail-oriented, focus those kinds of questions on QA candidates and other people for whom being detail-oriented is a critical part of their position.

For example, you can still be an excellent software engineer even if you aren't detail-oriented. For those candidates, you would be better off asking how they approach problems or collaborate with a team.

Although it takes extra time to customize your interview questions for every candidate based on the role, it will help you make better, long-lasting placements and hires. And that is what excellent recruiting is all about.

### Technical Roles Cheat Sheet

When it comes to tech hiring, many positions overlap one another. This [Technical Roles Cheat Sheet](#) is a recruiters' guide to some of the more common languages, frameworks and tools. It provides a baseline for matching up keywords on resumes and job descriptions to different tech disciplines. It isn't fully comprehensive, and different organizations may have different needs. However, if a candidate fits into one role, they will likely be able to learn the other skills listed.

### Conclusion

Since every role is different, and each candidate unique, the best thing you can do is to be great at understanding those differences and communicating them effectively. By doing the work many other recruiters won't be willing to do, you'll increase your ability to make outstanding placements and hires. That deep understanding of the roles, technologies and teams will help you make the best matches with the best candidates.

#### ABOUT THE Author



**Kate Matsudaira** is an experienced CTO and has spent the last seven years immersed in startups in architect and CTO roles. Prior to joining the startup world, she spent time as a software engineer and a technical lead/manager at Amazon and Microsoft. Kate has a passion for big data and all the challenges that come with it. She has lot of experience building large scale distributed web systems, with some mobile, cloud computing and engineering leadership thrown in for good measure. She maintains a blog at [katemats.com](http://katemats.com). [@katemats](#)



## Software Engineer

This is the most common job title, and it can mean all sorts of things. Generally, these include: writing code, fixing bugs, writing tests for code, and participating in design discussions (usually technical design, but in smaller teams or senior roles it could be product design).

If the engineer has worked on a large software application, it is possible that they may be specialized in a particular area of that application: the back-end, middle, or front.

### Questions for Software Engineers:

**Q What part of the system do you prefer to work in? Front, middle, or back? Why?**

**A** If someone is inexperienced or hasn't worked on a sophisticated system, they may give a generic answer or tell you they can work in any part. While that may be true of some folks, most experienced engineers will have a preference and likely a strength in one.

**Q What are the components of a great software development process?**

**A** Here, look for someone to mention things like product and technology definition, source control, build systems (continuous integration), deployments, along with testing, other quality measures, or documentation. There isn't a right answer, but experienced engineers will definitely have opinions. If you aren't sure of the response, ask about processes they didn't care for and how they would have made them better.

**Q What is the best way for people to collaborate on a complex coding project?**

**A** There are a lot of challenges when it comes to having two people work on the same piece of software. There can be conflicts in implementations (both people modifying the same code) or dependency issues (one person needs something from the other, but it hasn't been built yet or needs to be built in tandem). Good candidates will have ideas and suggest things like clear definitions, lots of communication and pre-established interfaces.



## Back-End Engineer

While the word "back-end" is certainly amusing in a job title, it describes the role well. These engineers typically work in what might be the back room of a store, but for software. The work they do (namely, the code they write) doesn't touch the users or customers; it isn't the user interface or part of the software you can see. More often, back-end engineers are writing code in the form of APIs or interfaces that then get consumed by others (such as middle or front-end engineers).

These engineers typically interface with the data (stored in files or in databases) or APIs to transform it and make it usable by other parts of the system. These engineers are often closest to the systems, web services (such as in the cloud), or hardware.

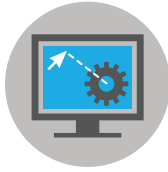
### Questions for Back-End Engineers:

**Q What are some strategies that connect the back-end of a system to the front-end?**

**A** This question has no "right" answer, but look for a sensible, clear response. For example, a valid answer might be APIs. Another person might say callbacks. If you don't know what these terms mean, ask for an example or analogy. This way you can see their communication skills and get your head around their solution. Then ask about pros and cons of their solution, and if they have seen any good or bad examples in their careers. Most candidates will have some thoughts and ideas, and even examples from their own experience.

**Q How do you choose which database to use for a project?**

**A** Don't be surprised if a candidate says it depends on the project. If that is the case, you can either give them an example, such as a website you like and use, or you can ask them for different criteria and examples. If they tell you something you don't quite understand, ask them to expand. The real key is that they are thinking critically about how the data will be used and consumed by the customer in the website or application.



## Middle-Tier Engineer

Middle-tier engineer roles are less common (it is much more common to have back-end engineers that work in the middle tier), but can be found in larger companies or teams working with large-scale software. Like back-end engineers, they also don't work on the user interface and instead are focused on being the glue between the back-end and front-end parts of the system.

Typically middle-tier engineers are responsible for the core of the business logic. If software were a grocery store, back-end engineers would get the stock off the trucks, and middle-tier engineers would make sure that the right amount of items were put in the right place. That middle-end logic makes it not just a room full of groceries, but a system of groceries organized to make a proper grocery store.

Generally the skills required for a middle-tier engineer overlap most with a back-end engineer, but the back-end engineer may have a bit more responsibility for the system or data storage of an application.

### Questions for Middle-Tier Engineers:

**Q** In your experience, what have been the responsibilities that separated back, middle, and front?

**A** By diving into their own experience, you gain insight into how work was separated, and how they interfaced with other workers. This is also a great way to understand if they veer more toward a back-end or front-end role, since most people in middle roles tend to lean one way or the other. Expand your questions to see how they worked with other engineers and collaborated on projects.

**Q** What is a cache? What are some examples of caches you have used?

**A** This is a great technical question that is easy for recruiters to ask. A cache is simply a place to store data that is faster to retrieve than the source. For example, your fridge is a cache for the goods at the grocery store. You can't store as much, but it is much faster to retrieve items from the fridge than from the grocery store. Hopefully candidates give you their own examples so the more times you ask this, the better you understand it.



## Front-End Engineer

Using our grocery store analogy, front-end engineers are like the cashiers serving the customers. They work on the most visible pieces of the system, often called the presentation layer. This role requires good communication with designers and product people since they translate the overall vision into something people can actually see.

Depending on the software being built, the front-end engineer can also be considered a User Interface (UI) Engineer, or if the software is web-based, a Web Developer.

### Questions for Front-End Engineers:

**Q** Explain MVC in simple terms.

**A** MVC is a programming methodology that is widely used for separating logic and business concerns to make implementation easier. **M (Model)** is the part of the code that represents data, or objects. **V (View)** is the part of the code that shows the things inside the Model. The actual views

of the data composing the user interface. **C (Controller)** is the part of the code that gets commands from the user and tells the View what to show, gluing the Model and View together.

To use an analogy, MVC is like a tv. The View is what you see on the screen. Your cable provider is the Model supplying the data and the channels, and your remote control changing what is in the View is the Controller.

**Q** What are some ways to optimize website assets or resources?

**A** There are lots of examples but a few right answers include: file concatenation, file minification, leveraging a content delivery network (CDN), caching, gzip or other compression strategies, reducing image resolution or sizes, creating image sprites, etc. You can also ask about challenges on mobile versus standard web performance. Good candidates will have lots of ideas and suggestions and be able to break things down into analogies or examples to explain how they work. If it sounds like they might be reading the answers from a document, ask for specific examples from their experience. Good candidates will be able to provide a lot more detail.



## Web Developer

Also sometimes called a User Interface (UI) developer or User Interface (UI) engineer in larger teams, the primary focus of this role is often the front-end part of a website. In smaller teams or simple applications, the role can encompass all aspects of the website (including responsibilities that might otherwise fall to back-end or middle-tier roles). Generally though, this role is focused on making a website look and behave as dictated by the design.

People in this role typically have a handle on client-side technologies like HTML, CSS, and JavaScript. The role can also include other presentation layer technologies such as Ruby (usually Rails, which is a framework built using the programming language Ruby), Django, and PHP. Many web developers have light graphic design skills and are more than capable of manipulating images and graphics using tools like Adobe Photoshop or Illustrator.

It is possible to encounter web developers that don't work in the details of the presentation layer, or are less skilled in layout (such as HTML and CSS). In that case, they would

have a strong grasp of other client-side technologies, like JavaScript, as well as strong middle-tier engineer skills.

### Questions for Web Developers:

**Q Can you describe your workflow when you create a web page?**

**A** When it comes to creating webpages, most developers break things up into pieces. They should be able to tell you how much “design” they need to get started (comps, wireframes, etc.) but mostly you are listening for thoughtful reasoning and planning.

**Q What is responsive design, and have you ever implemented any mobile specific layouts?**

**A** Responsive design is about creating sites to provide an optimal viewing experience depending on the client. All web developer candidates should understand the concept, although not all will have utilized it. For those who have experience with responsive design, ask about the techniques they have used. It is likely they will mention flexible or fluid grids, media queries, or flexible images. Another great follow-up is to ask about the challenges with responsive images, since there is no “right” way and there are some good pros and cons with each.



## Database Administrator (DBA)

This role is all about data, and more specifically, databases. A DBA may be responsible for database and data schema design (which can be critical in large scale computing environments) as well as database installation and maintenance. Many of these roles also encompass an operational piece, which can include simpler tasks like setting up servers and maintaining backups or more complex tasks, like performance tuning.

Unlike many other technical roles, this is one where certifications actually matter, or at least help to pin down a DBA candidate's expertise. While many skills translate across technologies, most DBAs tend to specialize in one particular type of database (e.g. Oracle, Microsoft SQL Server, MySQL). However, it is possible to have a DBA generalist that can navigate several different types of databases.

Typically, you don't see many DBAs specializing in NoSQL-type databases (most of which don't have specific certifications),

since those technologies are still very new and require big-picture system knowledge. In most organizations, you see NoSQL databases administered by software engineers or DevOps types.

### Questions for DBAs:

**Q When it comes to creating a new schema for a database, what is your process? In your experience, what works well and what doesn't?**

**A** Look for a thoughtful process, ideally one that focuses on really understanding the requirements of the users. It is especially useful to drill into their past experiences designing schemas and working with others to bring that data to life.

**Q What is a database?**

**A** This is a basic question, but it can be a great place to start because “database” is such a generic term. You can even ask them about the different types of databases and the criteria for choosing one over another. Hopefully they have good ideas and can break things down in a way that makes it easy for you to understand.





## DevOps Engineer

This is a newer role and represents the bridge between the worlds of building software and the operations required

to support it. As more organizations move into the cloud, and running systems becomes more about writing code than about configuring hardware, you'll see more of these roles and responsibilities being created.

This role requires a candidate who understands the operational best practices of a website, but who also has the programming skills (or at a minimum strong scripting skills) of a software engineer. It is very hard for someone to fill this role without at least some experience running a production website, unless the role is in a bigger company with a larger support staff to properly train new hires.

Software engineers are usually great fits for DevOps positions; excellent candidates will have a great passion for debugging and troubleshooting systems, and deeper systems knowledge. DevOps candidates can also come from system administrator roles who have a strong software and coding background, but with a solid handle on the operations required to deploy and release software. In general, this role usually requires a decent amount of programming, so it is more common to see an engineer moving into a DevOps role than a system administrator, because of the programming knowledge required going in.

### Questions for DevOps Engineers:

**Q** Have you ever been on-call? Do you mind it? Why or why not?

**A** First and foremost, look for someone with some basic operations experience. Most operational engineers can tell you stories until the cows come home. Ask about the worst time they were ever on call. What happened? How did they get things resolved? Great candidates will give you confidence in their abilities.

**Q** What are some attributes of a great post-mortem meeting?

**A** Post-mortems are key in a DevOps role, since they are the meetings that usually follow an incident or issue, and are focused on understanding what went wrong and how to prevent it in the future. Most operational engineers will have opinions be able to offer format and agendas that would make sense.

**Q** How do you know what to monitor in a system?

**A** Some engineers will tell you that you can never have too much monitoring. However, there are always metrics that matter more than others. For example, up time (is the website up or not?) is one of the most important, along with number of transactions per second, and system health (memory usage, central processing unit [CPU], input output processors [IOPs], etc.). As they rattle off ideas, ask why and how frequently these metrics should be watched. Great candidates will have tons of examples.



## System Administrator

System administrators, often referred to as sysadmins, are in the operational side software. They help set up servers

and hardware for the software to run on, and they make sure things stay up and running. This can include being on-call to be alerted when there is a problem with the system.

Many system administrators specialize in specific areas or technologies. For example, one might be an expert in mail servers (called a postmaster), security, or even internal IT (making sure your phones and internet work

as expected at the office, or removing the virus from that email you knew you shouldn't have clicked on). Some of these specialties have training and certificates, and while there aren't necessarily degrees in system administration, many trade schools and information technology programs specialize in this type of role.

If you are hiring for this role, it is really important to understand the responsibilities this specific company will need covered, since this position can be so varied. A person required to monitor and operate systems needs to be willing to be on call, and should have a strong track record of being accountable and responsible. Whereas a person in an internal IT support role should have customer service skills.

*continued on page 8*



*System Administrator, continued from page 7*

Some of these roles are expected to do light coding or scripting (which is a lot like programming, but with one-off pieces of code, versus more traditional programming, which is like building pieces into a larger puzzle).

**Questions for System Administrators:**

**Q** What is an example of a script you wrote?

**A** Make sure they have written a script at some point (even most good entry-level candidates have written a script before). Ask what it did and why they wrote it. Did they share it with anyone? If they were to share

it, what would they change? Insightful changes might be adding comments or documentation, or making it environment agnostic.

**Q** Someone tells you his or her computer is slow. How do you diagnose their problem?

**A** This question has no “right” answer, but look for basic problem-solving skills. What sort of questions do they ask? How do they diagnose the problem? If they mention a step you’re unfamiliar with, ask them to expand so you can see how they would work with a real customer that may have a similar issue.



## Network Administrator

Network administrators, like their system counterparts, are also in the operations and support side of an organization. Network admins don’t typically get involved directly with user or customers, but work closely with sysadmins and other engineers. This role’s main focus is on the network components of the technology and software infrastructure.

It is unlikely to see network admins in an organization that has a cloud platform setup, since their main responsibility is to make sure that hardware is set up correctly and that important systems have enough bandwidth and redundancy to serve requests quickly. In other words, this role involves understanding and setting up hardware and the topology of how that hardware is connected. In the cloud, all of that hardware is obscured, which is why you don’t see this role at cloud-based companies.

This role also has some certifications for different types of hardware, like configuring routers and switches, but many of these are not required since most network administrators don’t specialize in one type of networking

technology. In smaller organizations, this role can also encompass sysadmin responsibilities, like setting up and maintaining printers, firewalls, WANs, IP phones, and mobile networks.

**Questions for Network Administrators:**

**Q** What is a network?

**A** It never hurts to start with the basics. If the answer is satisfactory, follow up with “what is a router?” A router is like the air traffic control at the airport; it routes planes to the right terminal.

**Q** How does DNS work?

**A** DNS is core to how the Internet works. It stands for Domain Name System and is the primary mechanism that makes sure a web address (like [www.dice.com](http://www.dice.com)) ends up in the right place so that the site shows up in the web browser.

**Q** What are some important considerations when designing an office network?

**A** A good candidate will ask questions and mention things like making sure everyone has the resources they need to do their job. Ask about past experiences and mistakes, and what they learned from those challenges.



## Data Scientist

Data scientist is not a new role. However, with computing resources becoming more affordable and technology advancing more rapidly, more and more organizations are employing people in these roles to help them understand their data.

A major goal of data science is to make it easier for others to discern meaningful conclusions from data sets. And depending on the organization, the ‘customers’ can change. Sometimes data scientists are responsible for R&D, trying to research new business opportunities. Other times they can be a part of marketing teams (usually called “analysts”) and are responsible for providing insights into strategy, such as product positioning or segmenting existing customers.

This role can require a versatile skillset, depending on the needs of the organization. Some roles require a thorough understanding of the challenges of processing very large data sets (i.e. “Big Data”). Others require practical applications of exploratory data analysis, statistics, machine learning, and data visualization. Data scientists may have computer science backgrounds, but they can also come from math or analyst backgrounds.

When hiring for this role, be sure to understand the amount of programming required (and in what languages), since coding abilities can vary among candidates, and may not always be necessary to be successful.

Data scientists can also be more specialized, depending on the needs of the organization or a candidate’s background. For example: data miners specialize in mining data looking

for patterns or insights, and machine learning specialists focus on building and training models for repeatable analysis like artificial intelligence.

### Questions for Data Scientists:

**Q** Tell me a success story from one of your data analysis projects.

**A** Try to understand what made the project a success. Ask about the size of the data set, and how long they had to analyze it. What were the results, and how were they used? Then, ask about a project that was less successful and the surrounding details.

**Q** How do you know when your results are good enough?

**A** One of the challenges with data science is that there isn’t necessarily a right answer. Sometimes there are lots of ways to solve a problem, and new algorithms and approaches can make results slightly better. However, it is hard to verify whether those results are correct, so ask for examples of how to verify correctness in their findings. Some strategies involve creating golden sets of data that you know are definitely correct and testing your algorithm against those, or sampling some of the data and verifying those results are correct.

**Q** What types of clients or customers have you worked with in the past? How long did your projects normally take?

**A** Because these roles can vary so much, make sure the nature of their experience translates well to the position. You can also ask about working on a team and collaborating to share their results with others.



## Quality Assurance Engineer

QA engineers are almost always on the software-building side of the organization, and are responsible for making sure that software meets the quality bar. Their responsibilities include testing software by using it (like you do when you visit a website and click around), and writing programs to automate various tasks or tests for the system.

Some of these roles may require very little coding or scripting knowledge, and others may require a lot – it really depends on the organization and software they are testing. Regardless, strong attention to detail is important. Many QA engineers are responsible for coming up with test plans and test cases (imagine a big list of everything you would need to test to make sure something worked), as well as executing these plans.

People in this role should know how to break things, and be rigorous and thorough. This role tends to involve a lot of communication with engineers, designs, and product

*continued on page 10*

Quality Assurance Engineer, continued from page 9

people to verify functionality, so they should also be a good team player with strong written and verbal skills.

**Questions for Quality Assurance Engineers:**

**Q** What is your process for creating a test plan?

**A** A good candidate will offer a thoughtful process. Then follow up by asking about the components of a good test plan. Feature definition and individual test cases are typical, but some test plans can include variants for operating systems, different browsers, and even different data examples.

**Q** What is an example of a great bug you have found?

**A** Dive into their example. Was the bug on the test plan? What made it great? Was it fixed in a timely manner? Ask lots of details about it and why they considered it an achievement.

**Q** What is regression testing?

**A** When bugs are fixed in software, the changes can have unintended consequences, including new bugs. Regression testing is making sure that new issues haven't arisen from a change that was made. To follow up, ask for an example of a regression they have seen and the conditions under which they found it.



## Software Engineer in Test

These roles are similar to QA engineers in that they tend to have a lot of the same responsibilities around test cases, planning, and quality testing. One key difference though, is that software development engineers in test (or SDETs) are also responsible for building and creating programs to help with testing and quality. Depending on the specifics of the role, this can mean more QA or more coding. The coding in the role is seldom on the product itself, but instead on other programs to help increase the quality of the product.

Sometimes SDETs can be responsible for developer tools – like a software build system code repository – or deployment tools that can help other engineers write better, higher quality code. Even though the title is different, a good SDET could easily be a software engineer; they just think differently or have a passion for developer productivity. This is different than a QA engineer, who usually doesn't have the programming chops required to succeed as a full software engineer.

**Questions for Software Engineers in Test:**

**Q** Why an SDET role over an SDE role?

**A** Theoretically the roles should be similar in skill around coding, but different in the way they think and what the candidate is passionate about. Asking why they love testing and what draws them to that role is an insightful way to understand the candidate's passion and what makes them tick.

**Q** Tell me about a tool you built. How was it used? What was the impact to the team?

**A** This role is often focused on building tools to support and improve their team and process, so it's important they think about their work in this context. How was the work defined? How long did it take?

**Q** Do automated testing tools make testing easier?

**A** Possibly. For small projects, the time needed to learn and implement them may not be worth it. However, for larger or ongoing projects, they can be valuable. Follow up by asking for examples of common test automation tools. These include code analyzers, code coverage, recorders, load and performance tools, and many more. You can also ask the candidate about what each tool does.



## Technical Lead

The many different flavors of this role can be called Development Lead, Technical Lead, Lead Software Engineer, Software

Engineer Lead, Software Manager, or Lead Developer. While the responsibilities differ from company to company, this position is usually held by someone with a software engineer or developer background, who is responsible for the work (and often the people management) of several team members.

Typically this person is senior and more experienced (in years, or in legacy system knowledge for that organization) than many of the other team members. They go to meetings and represent the technical team to other teams in the organization. This role may also include a project management role, working with the design and product teams to create requirements, or supervising system operations and quality (QA).

Technical leads are often responsible for assigning and delegating work among their team, and for reporting progress against deadlines. Many leads serve as technical

advisors to management and other business functions, acting as a bridge to the non-technical functions in the organization. This role requires strong technical skills, but also an aptitude for leadership and communication.

### Questions for Technical Leads:

**Q** How do you divvy up work to your team?

**A** There are no right answers, but they should mention balancing skill-sets with personal preferences, and what is needed from the business in terms of results. You can drill into the specifics, or you can ask how to handle assigning work items to someone that they know won't like the work. How do they break the news? Has this ever happened to them?

**Q** What is the best way to integrate and onboard people onto a technical team?

**A** Most good tech leads have experienced new hires starting at their company, or have been a new hire themselves. Ask what works well and what doesn't. Most good leads will mention mentors, regular checkpoints with clear goals, and/or starting with small work items like a bug and progressing to larger tasks.