

# How to Measure Your Tech Recruiting Success?

Track the right metrics and identify leaks in your recruitment funnel

# Table of Contents

- 01 What metrics to track?
- 02 Remind yourself of your goals
  - Grow your developer pipeline
  - Reduce tech recruitment spending
  - Generate more developer applications
  - Improve awareness of your company
- 03 Identify underperforming ads and bottlenecks in the hiring process.
  - Lessons from Marketing
- 04 Ten examples of key performance indicators you can track and action items for follow up

# What metrics to track?

In today's candidate driven market— with a global shortage in tech talent—there are probably only a few companies that can say “We are doing fine attracting plenty of qualified candidates.” Even for firms with a healthy pipeline, there is no recruitment and employer branding team out there that does not constantly need to optimize and stay on its toes in order to continue attracting the right talent in a fast changing marketplace.

Since the assumption is everyone could always be doing more, the question becomes: more of what? Some recruiting and employer branding teams would not be able to say how to best invest, or which tactics are really worth their time. The question seems simple enough, but digging for the right answer is crucial.

## How is your tech recruiting strategy performing?

Measuring the ROI of your recruiting efforts is no easy task. Which KPIs (Key Performance Indicators) are meaningful? Well, it's something many people have puzzled over. One of them, [Wayne W. Eckerson](#), argues good recruiting is “as much art as science.” Still he goes on to list characteristics of likely high-impact KPIs:

- **Sparse:** Don't catch a case of analysis paralysis. Eckerson's first rule: “The fewer KPIs, the better.”
- **Drillable:** While every team member should be working towards a KPI, the high-level view on a dashboard should only contain a handful of strategic KPIs.
- **Simple:** Users understand the KPIs and know what is a good benchmark. For example, a recruiter who writes [20 good recruiting emails per week](#) is more productive than the one who writes 30 that fail to get any traction. The KPIs should reflect that.

- **Actionable:** Your team should know how to affect outcomes, but don't panic at the first red numbers. Especially if you are introducing KPIs for the first time. Eckerson's advice: "Managers need to manage according to overall trends rather than the current status of a KPI." For example, pay more attention to whether critical data points are improving quarter over quarter or year over year, as weekly results may bounce up and down.
- **Owned:** Each KPI should have one person looking out for it and informing the team on performance—successes as well as negative trends.
- **Referenced:** Establish on single source of truth. Clean up your data and train your team on the source to establish trust.
- **Correlated:** KPIs should drive desired outcomes and impact business goals. So it makes sense to evaluate them frequently. Some KPIs might even have a limited lifespan by design. Eckerson points out: "It's important to correlate KPIs on a continuous basis because their impact changes over time as the internal, economic, and competitive landscape shifts."
- **Balanced:** KPIs should consist of both financial and non-financial metrics.
- **Aligned:** KPIs should not steal resources from each other. But Eckerson also points out that this is hard to predict. "Many performance management practitioners caution against striving for KPI perfection. They recommend getting KPIs to a point where you are 80% confident they will work and then committing to practice to see what happens."

Another popular definition to think about are 'S.M.A.R.T. goals.' The term first appeared in a 1981 paper by George T. Doran titled "There's a S.M.A.R.T. Way to Write Management's Goals and Objectives." Doran wrote that goals that drive projects and businesses forward answer the following questions:

- **Specific:** What is a particular area in which you'd like to improve?
- **Measurable:** How will you measure and quantify progress?
- **Attainable:** Who is responsible for executing each task?
- **Relevant:** Does this goal help our entire company move forward?
- **Time-Bound:** When do you think this can be achieved?

## How can these translate into the tech recruiting world?

- **Specific:** Your tech recruitment goals should be as specific as possible. Instead of saying “We need more candidates,” think about the specific roles and ask: Do you want more developer applications in general? Do you want to improve your recruitment email response rates? Or do you want more candidates with a certain skill set or level of experience?
- **Measurable:** For each goal you set, you should be able to measure success along the way. Let’s say you currently generate ten developer applications per month. Without a clear goal, you’d technically hit this target by generating just 11 the following month. Let’s tweak this a bit. “We want to increase developer applications for our mobile developer role by 20%.” That’s not only a more powerful statement, but it also enables you to track your progress along the way.
- **Attainable:** You might have aspirations of generating hundreds of developer applications by the end of the quarter, but take a step back and evaluate what’s realistic. Factors such as your current headcount, responsibilities beyond tech recruitment, and budget all play a role in defining each S.M.A.R.T. goal.
- **Relevant:** Each S.M.A.R.T. goal that you create should answer the following question: “Does this help our company move forward?” If not, there’s no point in allocating time and resources towards hitting it. Hiring four Full-Stack Developers is cause for celebration, but not if your engineering lead really wanted to add four new Mobile Developers.
- **Time-Bound:** Each goal should have a timeline. If you’re going to improve your tech recruitment email response rates by 5%, you need to know when you can realistically achieve this target. Why? Because once you’ve hit this goal, you can move on to other objectives, such as growing your candidate pipeline and reducing your cost-per-hire

# Starting with some top level goals

You may have found Eckerson's tips or the S.M.A.R.T metric more useful, but ultimately it doesn't really matter. You should start by brainstorming the things you really would like to know about your tech recruiting efforts and their impact, and then try and turn those into actionable metrics to work towards. Our recruiting experts talk to companies everyday, offering advice on how to attract, engage, and ultimately hire developers. Here are some of the pain points they hear from clients most often.

## Some of the biggest pain points of recruiters in the tech space today:

- We need to find more (quality) candidates
- We need to find talent with a limited recruiting budget
- We are not getting good applications
- We are not known among the tech audience

This list is by no means exhaustive, so add what makes sense for your company. But just to get your head around how to start measuring, let's look at these examples in more detail and translate needs into strategic goals.

### We need to find more (quality) candidates

#### >> Grow the developer pipeline

For this one, you will probably need to drill down into which open roles you are looking to fill, then repeat the exercise for all of them. Remember the top level-dashboard Eckerson was talking about? This will be one of the examples where "more developers" isn't specific enough, and you would be well served to explore in more detail how open roles connect to what different teams specifically need.

Let's say, for instance, engineering leadership has told you to anticipate a future need for Database Administrators. Now the next step would be to create tactics to increase that number of 'warm' leads in your network. You might define "Attend x amount of database-specific technology meetups per quarter" as a goal. You might then monitor the metric of how many passive candidates your dev team and recruiters meet, with an overall goal to increase your network of candidates by x% by a certain date.

## **We need to find talent with a limited recruiting budget**

### **>> Reduce tech recruitment spending**

For many companies it often feels like the spend for recruiting is increasing but the number of hires is decreasing. With many platforms and job boards running at the same time, there are many factors at play. The metric you might want to track is the cost-per-hire for each developer role.

Here some planning arrives at the catch 22: we need to move away from ad-hoc spending on recruiting agencies, but we don't have the money to do anything more strategic, because we spent it all on those much needed hires. No surprise, your metric needs to be about figuring out which are your most profitable channels for quality candidates. So start by getting feedback on the candidates that impressed your team the most and figure out how they arrived in your hiring pipeline. At the very least, ask all candidates who make it to the screening interview stage, where they have seen your ads.

## **We are not getting good applications**

### **>> Generate more developer applications**

Similar to the problem of recruitment channels, it's never easy to pin down all the effects one job ad has. Sure, you should A/B test different job titles in ads, but it's not always as clean as just a click-through-rate on your job application portal. Many of the touch points are softer and add up: A developer who saw a talk by one of your tech employees might remember it fondly when seeing your banner ad somewhere, then apply a few days later when they have some free time.

On Stack Overflow, you can advertise your job listings and employer brand in the context of a developer's daily work. As the developer's "second screen," Stack Overflow Talent consistently gets you in front of developers at the moment they are searching for tools, answers, and knowledge. But this also means they might see your job ad at their current job. Instead of hitting apply straight away, they might navigate to your careers page over the weekend. Stack Overflow can identify when users navigate to your desired page within 30 days of seeing the ad. When factoring in influenced applications, Stack Overflow ads drive six to nine times higher conversions than clicks alone. So this serves as a reminder that many touch points all factor into your ROI.

### Note

One of the overlooked metrics when you are looking to increase developer applications is the size of the candidate pool. You might be the best at attracting devs within a given bucket. But if you are recruiting for a Amsterdam head office and you are only open to people from the Netherlands, your pool might just be too small. In those cases, look into the benefits of remote work or increase your relocation packages.

## We are not known among the tech audience

### >> Improve awareness of your company

In the early days of establishing your company as an attractive employer, it can be hard to know which strategy you should begin with. Think about it this way, developers want to have an idea of what it's like to work at your company before they spend their time interviewing. Now your job is to find the most cost effective ways to make that happen. Start by asking: Can the relevant information even be found online?

It sounds simple enough, but it's astonishing how many companies have lackluster Careers/Work Here pages on their website. Aside from listing out your open positions and office locations, consider adding the following:

- Employee testimonials (in any format – text or video work well)
- Company mission statement and values
- List of benefits (tech-specific ones, too)
- Clear photos of the office space/desk setup of employees
- Sections for each team (potentially with photos or a description of each)

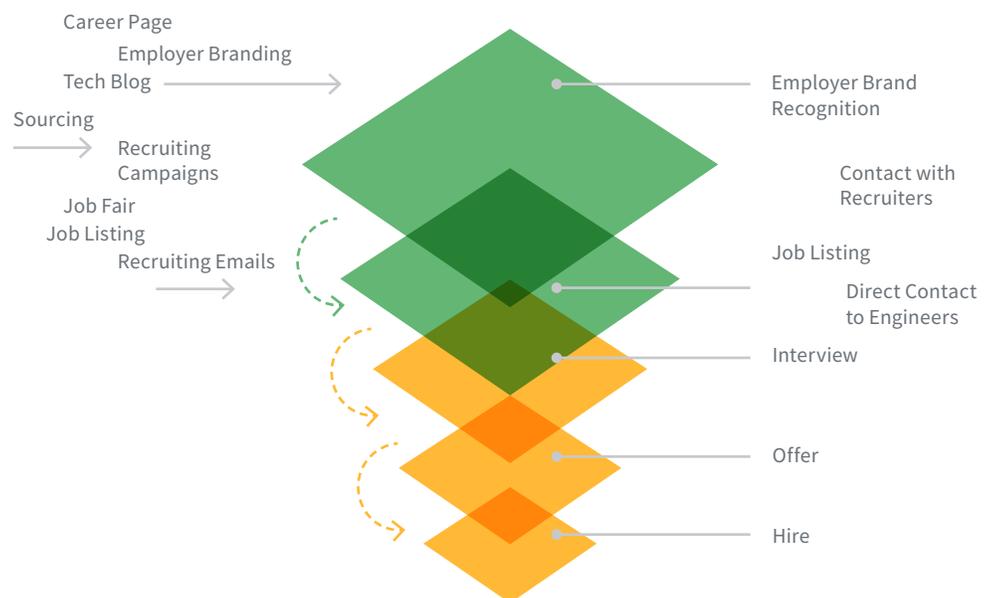
If a redesign of your entire careers page seems daunting, try and find a quick fix, like a “Work here” series of blog posts on a tech recruiting blog. On Stack Overflow Talent, companies can create a Company Pages to host a wealth of technical and editorial content. Get inspired by some of our [favorite examples](#).

An easy way to get enough content is to rely on your current tech team. This developer-generated content can include photos, projects, or testimonials from them. Ask your developers, “If you were job hunting and came to our profile, what would you want to see?” KPIs here can be Create x amount of developer artifacts (talks, videos, slideshares, conference write-ups, or other blog posts). If you have developers with followings on social media, recruit their help to grow the social media following of your company by x%.

# Find underperforming ads and bottlenecks in the hiring process

After these initial musings on the most common pain points, you might still ask, ‘But which KPIs to measure?’ It’s about identifying the things that already work well, but in most companies, there are also a few gaps in the communications strategy, and finding those weak spots can help you maximize your ROI. Let’s take a step back and look at an analogy very common in marketing: the funnel. Also called ‘Conversion funnel’, it looks at how many customers move from one level to the next.

## Lessons from Marketing: The Candidate Funnel



If your company is established enough, the candidate’s process usually starts with a detailed research of your company. This is followed by touch points with recruiters, employer marketing content, employer evaluation portals, regional press, company employees, and of course classic recruiting campaigns with job advertisements. The sum of these parts ultimately determine whether a developer applies for a job ad or replies to your in-house recruiter’s email.

Further down in the filter are those candidates who have decided to apply, have an interview, and finally decide on taking the job. In addition to classic recruiting work, the Candidate Experience matters for future hires. Learn from those new employees what they liked and disliked about your recruiting process. Discovering which parts of your funnel gave candidates pause provides valuable feedback you can use to iterate and improve.



# Ten example KPIs and possible action items

Let's look at some specific KPIs and possible action points. The KPIs discussed using the funnel can answer the questions that are urgent right now. For example: How do I split my budget across recruiting platforms? However, measuring and tracking some metrics will also reveal new questions.

In which areas do high indirect costs arise? Where and why do you lose an unusually large number of candidates in the application process? How can you generate more traffic for a job advertisement?

**Which alternative channels could play a role?**

What metric to look at	Why this might be insightful
Time-to-fill vs Time-to-hire	Both start from the time your job listing goes live. Time-to-hire is the time you need as a team to arrive with a candidate to sign a contract. Time-to-fill refers to the actual start date. This might not be due to a bottleneck in your process; there are also regional differences. For example, <u>UK, Ireland, and Europe on average take the longest to fill a position.</u>
Interviews per hire	This can be interesting to look at to see if the recruiting team nailed the profile (and their pitch to candidates). It could also mean that your time-to-offer is too long, so interested candidates sign-on with nimbler competition instead.

### Ratio of Views to Applications

Let's say that you've only had five applications for an open developer role. Pretty alarming, right? In some cases, this might be a sign that your job listing isn't working. But in others, it could actually be an indication that they're attracting only the most relevant candidates.

If views and applications to your job listings are both down, it's probably a sign your job listings aren't top notch. But if a high percentage of your total views are translating into applications, the odds are that you'll need to think of new ways to advertise your job listings rather than rewriting.

---

### Separate metrics for individual roles

This is where the “drillability” we mentioned in the opening chapter is most applicable. Make sure you know which job openings are working, and on which channels. This will allow you to test the hypothesis:

“Does this role need a better job description?”

“Do our DevOps Jobs actually perform better on Severfault.com than Stackoverflow.com?”

---

### Separate KPIs along the hiring process for different teams. (Even for identical role types)

“Maybe that Junior Dev position in one of your Android teams has a long time-to-hire because the hiring manager is slow to respond?”  
Different recruiting styles come into play here. These should be addressed for each team individually.



Seniority levels needed

As tempting as it is to think of a Junior and a Senior Dev with otherwise identical profile as basically the same person, the type of content, events, and recruiting emails that appeal to them are very different and their success should be tracked separately.

---

Alternative cost of open positions

This not only helps to prioritize several vacancies. Over time, opportunity costs can be used to make a case for a higher recruitment budget, since they might be much higher than what you are asking to spend to fill a role.

---

# of referrals over time

In addition to retention, this may be a good indicator of whether your own employees enjoy working for the company. From a [past developer survey](#) we know that 27% of developers found their current job through friends.

---

Feedback from Hiring Managers

Naturally, you will see a lot more candidates pass initial interviews than those who end up signing on. If the sample of candidates is small to start out with, you might go a long time without any hires. So make sure you have constant check-ins about the quality you pass on to your engineering team to make sure you are on track.



## Retention rate

If you not only want to make sure that many developers are hired as soon as possible but also want to know which ones stay, take a look at the percentage of employees remaining after 6 or 12 months.

This number should always be in the context of the team structure. If, for example, specialist departments were downsized, such a KPI was still not a reflection of the recruiting work.

It's important to remember that your employer brand should be much more than a recruitment pitch. It's a promise that you make to potential and current employees. If you're not living up to that promise, you'll struggle to attract and retain tech talent. In fact, declining retention rates could be a sign that your employer brand doesn't align with your developers' experience at your company.

Depending on your company's size and your department's roles, you may be sourcing, interviewing, and recruiting by yourself. You may also be creating and guiding the process for your subordinates to take action on. What success looks like on your team is something only you can define and evolve.



At Stack Overflow, we'll partner with you throughout the entire developer hiring process. From understanding the developer market to sourcing the right talent, we help you reach the right talent right now.

[Talk to an expert >](#)