Principles to stand out - straight
from the people running the
developer hiring process

THE TECH
RESUME
INSIDE OUT

What a good developer resume looks
like and how to write one

GERGELY OROSZ

With insights from hiring managers and
tech recruiters at well-known tech
companies

Sample Contents

The sample contains parts of Chapter 3, Chapter 5, and showcases the table of contents for The Tech Resume Inside Out.

Chapter 2: The Hiring Pipeline

Let's look at the bigger picture of what the interview process looks like, to better understand why a good resume is so important. This process can often seem like a black hole. It could also feel like a hard-to-predict series of interactions with people until you—hopefully—get an offer.

Hiring managers and recruiters look at this process quite differently and call it the Hiring Pipeline. Why this name? It's because at every stage, there's a significant dropoff in the number of candidates still in the pipeline.

People in the Recruitment Process

Throughout the hiring process, you'll interact with several people. Still, there are even more whom you might not be aware of. Let's take a look at each of the roles, their goals, and why you should care about these.

Hiring manager

Recruiter

Sourcer

Inbound sourcer

Recruitment coordinator

Interview panel
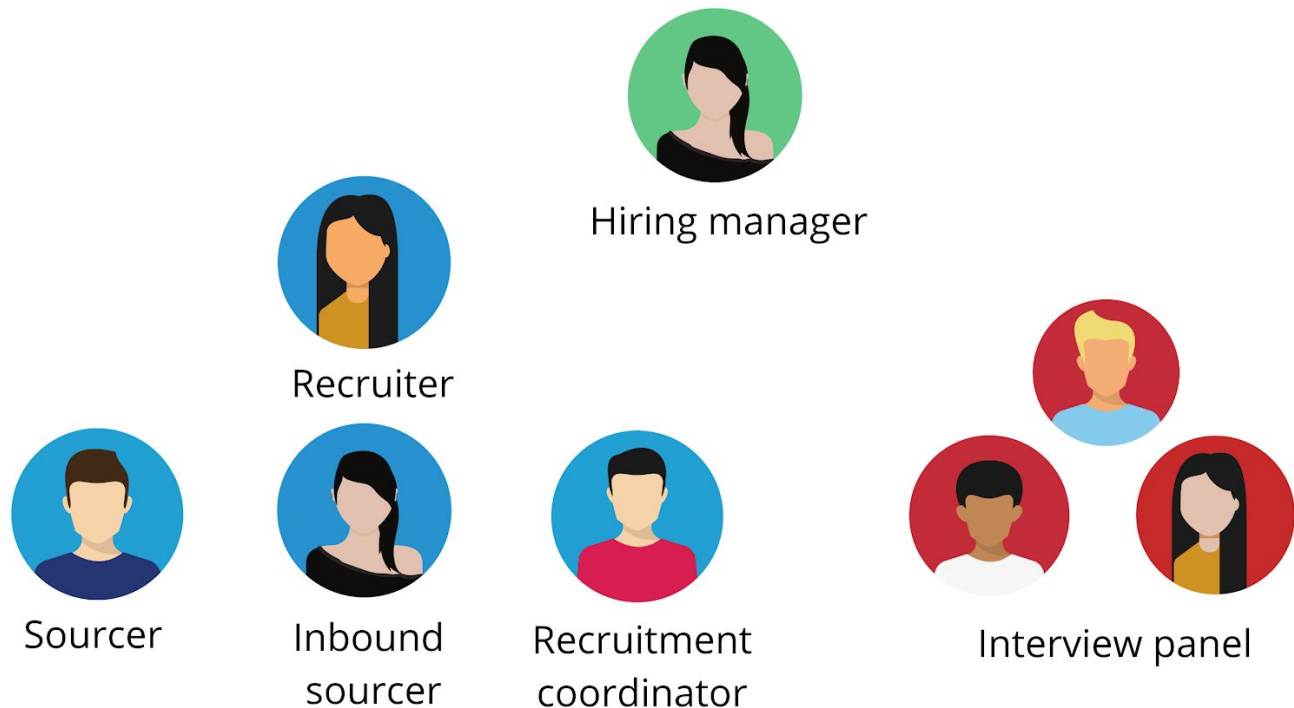
- **The hiring manager** is the most important person in the whole process, and they run the show. They are the person who has opened one or more positions—or headcounts, as it's internally called. They define the requirements they are looking for, and they usually write the job description. They set up the hiring process and define who the technical interviewers will be, and what areas they should focus on. They are the ones who have the final hire/no-hire decisions. Candidates usually don't talk to the hiring manager until they come onsite. The goal of the hiring manager is to hire people onto their team who will help this team excel.
- **The recruiter** coordinates everything on the recruitment side. From the point of a profile being promising, they are in touch with the candidate, guiding them through the interview process. There are several other recruitment responsibilities, which sometimes have dedicated people at

larger companies. The goal of the recruiter is to fill the roles that the hiring manager asks them to. To fill these roles, they need to find candidates who meet the bar set by the hiring manager and the interview panel. Recruiters often have target numbers to hit, measured in the number of headcounts filled.

- **The sourcer** proactively reaches out on LinkedIn and other channels to "source" people: to get them interested in starting the process. In large companies, sourcers exclusively do reachouts and sell the position via conversations. As soon as someone who has a good profile is interested, they hand this person over to the recruiter. The goal of the sourcer is to get as many qualified candidates in the pipeline as possible, and they usually have target numbers to hit.
- **The inbound sourcer** screens all incoming job applications through the company jobs site. This is a specialized role at larger companies, particularly Silicon Valley-based ones. At large companies, there could be hundreds of applications per week for each role. With tens or hundreds of roles, just going through these can take multiple people, full-time. Referrals will usually go either to inbound sourcers or to recruiters, with a priority over other applications. The goal of the inbound sourcer is to get qualified candidates forwarded to the recruiter. At the same time, they need to not waste the recruiter's time with people who don't meet the expectations set by the hiring manager.
- **The recruitment coordinator** manages the logistics of the process. Once you make it through the recruitment chat, and it's time for the technical phone screen or onsite, this is where they join in. They schedule times with you and with the interview panel. This can be more complex than you'd assume. If an interviewer cannot make it, they swap them out for a replacement. If something comes up on your end, they reschedule the interview for you. If you get invited to travel onsite, they take care of the logistics of booking transport and accommodation. The goal of the recruitment coordinator is to make sure things flow smoothly, and that everyone is happy and on time.
- **The interview panel** is the group of engineers who will lead the technical interviews at the technical onsite, from the coding challenge to the final onsite. They have usually gone through some training, have been calibrated, and the people usually specialize in doing specific interviews, like coding or design. The hiring manager selects this group. For small companies, this group will often be team members. For large companies, it can be a large pool of all engineers above a certain level who have taken the interview training. The goal of the interview panel members is to keep the hiring bar fair, consistent, and as bias-free as possible.

How does knowing about these roles help you in a job application process? It places things into perspective. For example, many candidates don't assume that both the recruiter and the inbound sourcer are fundamentally on their side. Both of these people need to make hires to hit their goals. However, they need to balance the expectations set by the hiring manager. If the hiring manager explicitly asked for certain technologies or certain years of experience, they will most likely follow this guidance.

Similarly, it's good to know that when you get a LinkedIn reach-out, it's from a sourcer. However encouraging you'll find things they say about your profile, these people want to get you in the hiring pipeline, and you don't have a job "guaranteed". You'll still have to go through the interview process.

When you interact with people, be mindful of these roles and their constraints. When a recruiter messages or calls you about a rejection, know that they are often a messenger. They are as invested in you getting the job as you are! As much as both the resume screening and the interview process can seem like a black box, it's run by people who try and do their best.

From the inside out: what can you typically expect from a recruiter?

Blake Stockman, who recruited for Google, Facebook, Uber, Flexport and other Silicon-Valley startups, explains how recruiters come in all shapes and sizes, and how your experiences can vary wildly between companies and recruiters.

The thing about the field of recruiting is that it's not one that people plan to get into. If you talk to anyone in the recruiting field, you'll find a wide variety of how people got here. This is in contrast to, for example, software engineering, where it typically takes a computer science degree, or some other technical degree to get started. No matter how you get into the software development field, it's rarely an accident: you plan and prepare for a very long time. With recruiting, people typically just find themselves in this industry.

As a result, you have a variety of backgrounds and expectations. You also find a varied level of quality with recruiters. This makes it hard to find that top tier recruiter who's going to help you really navigate that process well. Someone who will be your advocate, and someone who will give you a true view of what's going on.

What you should expect, and what the baseline should be to hope for, is a level of transparency of what's going on. To understand who this person's stakeholders are. For example, at a large company, like Google or Facebook, I would have a level of relationship with hiring managers, but it was really abstract. I was more of an administrator of the existing recruitment process. Here, all I could do was communicate to candidates what the process was, and what I was going to do on their behalf, as they made their way through the process.
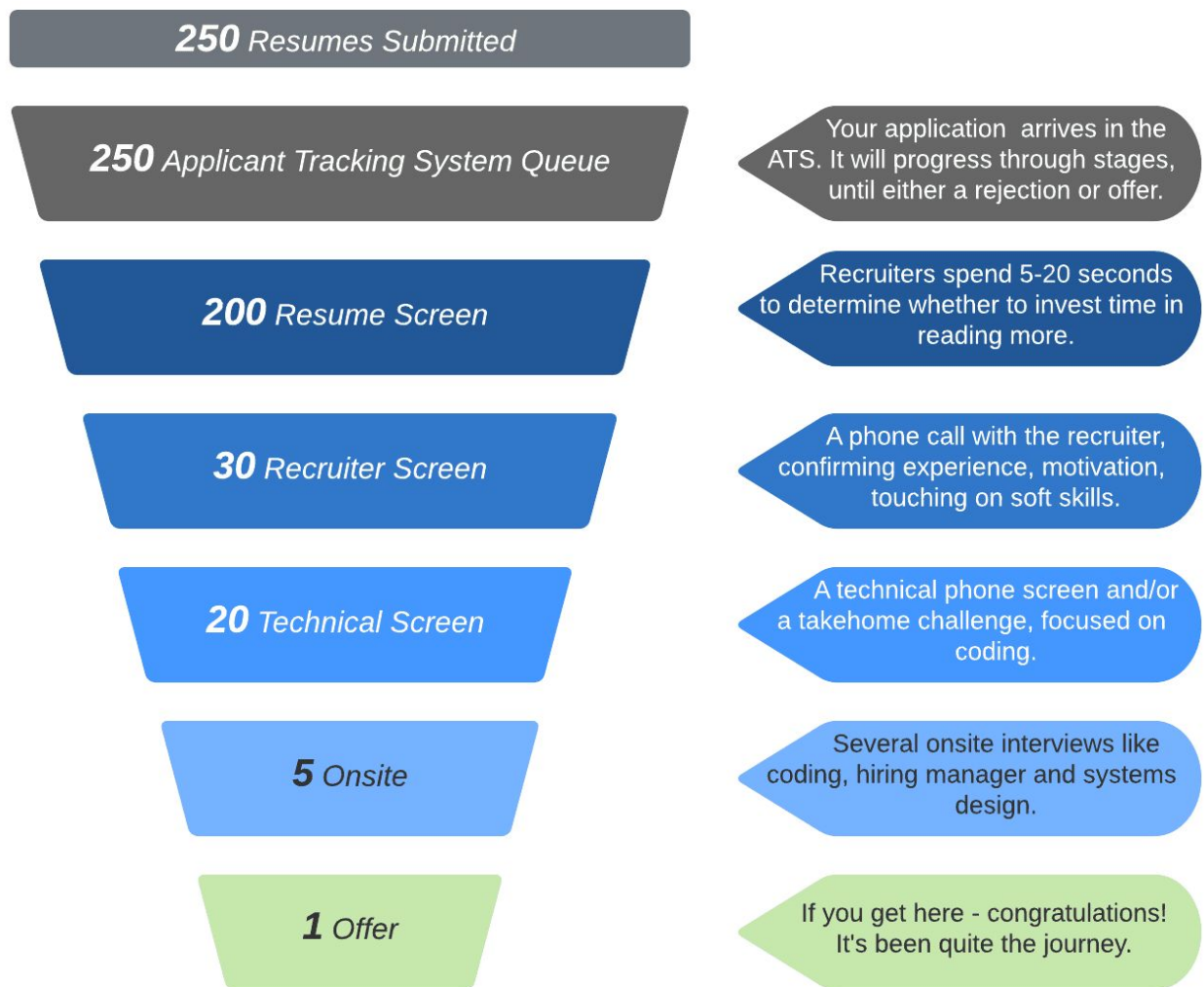
Whereas when I was at Uber, I was working with hiring managers directly on actually crafting what that recruitment process was, specifying the outputs of interviews. In this case, I was able to tell candidates what to expect at the level of focus areas—without giving away the exact questions themselves. Things like how there will be the coding, architecture, and hiring manager interview. How interviewers will want to know about your thought process, how outcome-oriented you are, how process- or thinking-oriented you are, and how they'll dive into collaborative problem solving with you.

Generally speaking, the smaller the company is, the more details you should be able to get, *and the deeper the relationship should be with your recruiter. So when you work with recruiters at startups, you'll get a much clearer view and a much deeper lens. You'll get a better idea of the culture, what's actually happening there, what teams are looking for, and what to expect.*

Of course, a lot depends on the actual recruiters. Some of the best recruiters I know are working at larger companies, and many of the ones at smaller companies are just getting their careers started. So you will have a high level of variance.

The Typical Hiring Pipeline

When you submit your resume through a job advert, the typical hiring pipeline is similar across all tech companies. There's a resume screen, a recruiter screen, a technical screen and a series of onsite interviews. You could get rejected at each round—or, if you did well, progress until you get an offer. Here is how this hiring pipeline could look, visualized:

250 Resumes Submitted

250 Applicant Tracking System Queue

Your application arrives in the ATS. It will progress through stages, until either a rejection or offer.

200 Resume Screen

Recruiters spend 5-20 seconds to determine whether to invest time in reading more.

30 Recruiter Screen

A phone call with the recruiter, confirming experience, motivation, touching on soft skills.

20 Technical Screen

A technical phone screen and/or a takehome challenge, focused on coding.

5 Onsite

Several onsite interviews like coding, hiring manager and systems design.

1 Offer

If you get here - congratulations! It's been quite the journey.

A typical hiring pipeline at a large, well-known tech company. The actual process, and numbers will vary on a role-by-role, company-by-company basis.

Let's look at each of the stages:

1. Applicant Tracking System Queue. Almost all tech companies use an Applicant Tracking Systems (ATS). These systems track the lifecycle of your application. For example, after a recruiter reviews the resume, the status of the candidate might change to "Reviewed" or "Resume Reviewed."

Not all applications might progress to the resume review screen. The reason some applications might not be looked at that there are already enough qualified resumes in the previous batch. Another reason could be that the position has just been filled. In general, the later your resume comes in, the higher the chance that it won't be looked at, due to having enough qualified candidates in further stages of the pipeline. With better companies, candidates would still get a standard rejection notification when their application is not processed further.

2. Resume / CV screen. In most companies, recruiters do the resume screening. However, some larger companies, particularly Silicon Valley-based ones, tend to have a specialized role called inbound sourcer, who takes on this screening. The recruiter or inbound sourcer scans your resume, spending 5-20 seconds to determine if you might be a match for the position. If you're not a good match, you might get an automated rejection message—assuming the company has a policy to send one. If the first scan is promising, they'll spend a bit more time reading your resume in depth. Afterwards, they might set up an initial call to check on a few details, and determine if they'll have you start the interview process.

The dropoff is usually the steepest at this step. From hundreds of resumes, only a few dozen profiles tend to have a recruiter screening call. This is why it's important to tailor your resume, so it grabs the attention of the recruiter, or the hiring manager at this point.

3. Recruiter screen. Only profiles who look like a promising fit for the job make it to the recruiter screen. This is a phone call with the recruiter, confirming experience, motivation, and touching on soft skills.

4. Technical screen. A technical phone screen and/or a take-home challenge focused on coding. Candidates almost always interact with software engineers at the company at this point.

5. Onsite interview. Several onsite interviews such as coding, hiring manager and systems design. The hiring manager interview usually looks for soft skills, culture match and values alignment. Systems design interviews are usually scheduled only for more experienced candidates.

6. Offer. It's common for a funnel that started with 100+ qualified resumes to result in a single offer. If you do get an offer, massive congratulations—you probably had little clue until now just how competitive this process was.
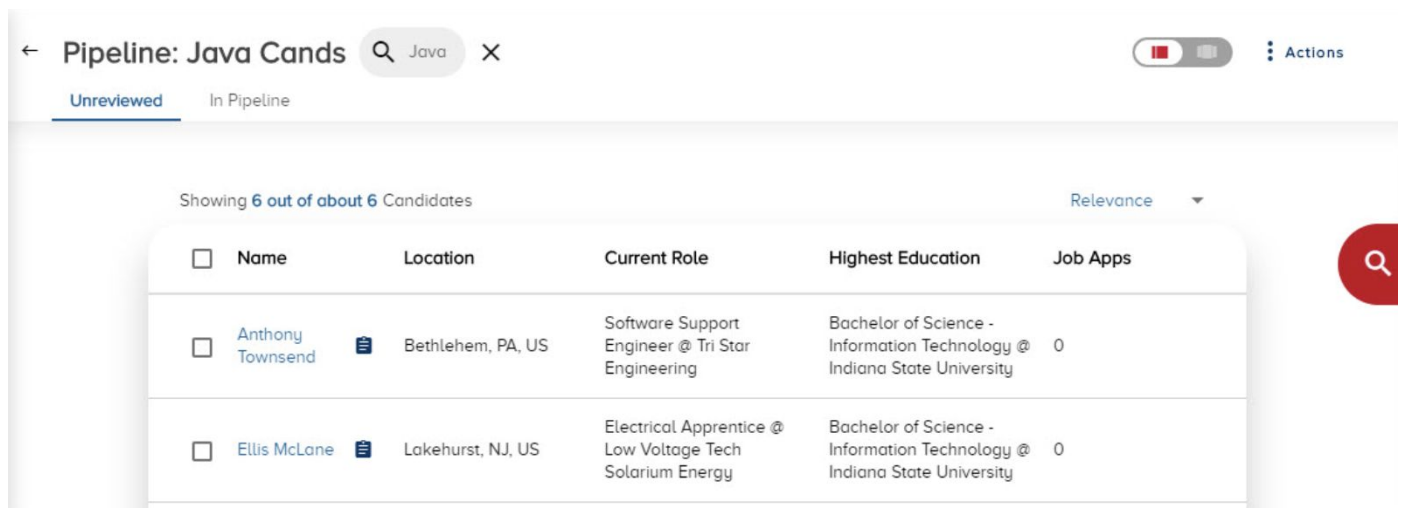
The Applicant Tracking System

Applicant Tracking Systems (ATSes) are a tool many candidates speculate about, and ask the question: do ATSes reject resumes? Spoiler: they do not. Humans do.

Some of the most popular ATS systems used by tech companies are iCIMS, Taleo, Greenhouse, Workable and Workday. There are more than 100 ATS systems, and more entering the market over time. While a few never, niche ATS systems are starting to offer more advanced functionality with resume parsing, as of 2020, none of the major ATS systems reject resumes automatically, or hide them from recruiters. Resume processing for all, major systems is basic. Most of the added ATS functionality is outside the resume processing phase, and with how these systems integrate with other company internal systems, and help with reporting.

The purpose of an ATS is to help coordinate the application process, for recruiters within the company. It keeps track of the status of applications, and exposes this information to all recruiters. For example, this is how a sourcer will know not to reach out to a candidate who is currently interviewing with the company, or someone who has recently declined an offer.

Here is how a popular ATS, iCIMS displays candidates in the hiring pipeline for a specific position —people who applied on the career page, who have been referred, or who have been sourced and added by a recruiter:

The pipeline view in the iCIMS ATS—one of the popular ATS systems. Recruiters need to review the candidate to process. This usually means them scanning the resume.

The ATS workflow is human-based and goes roughly like this:
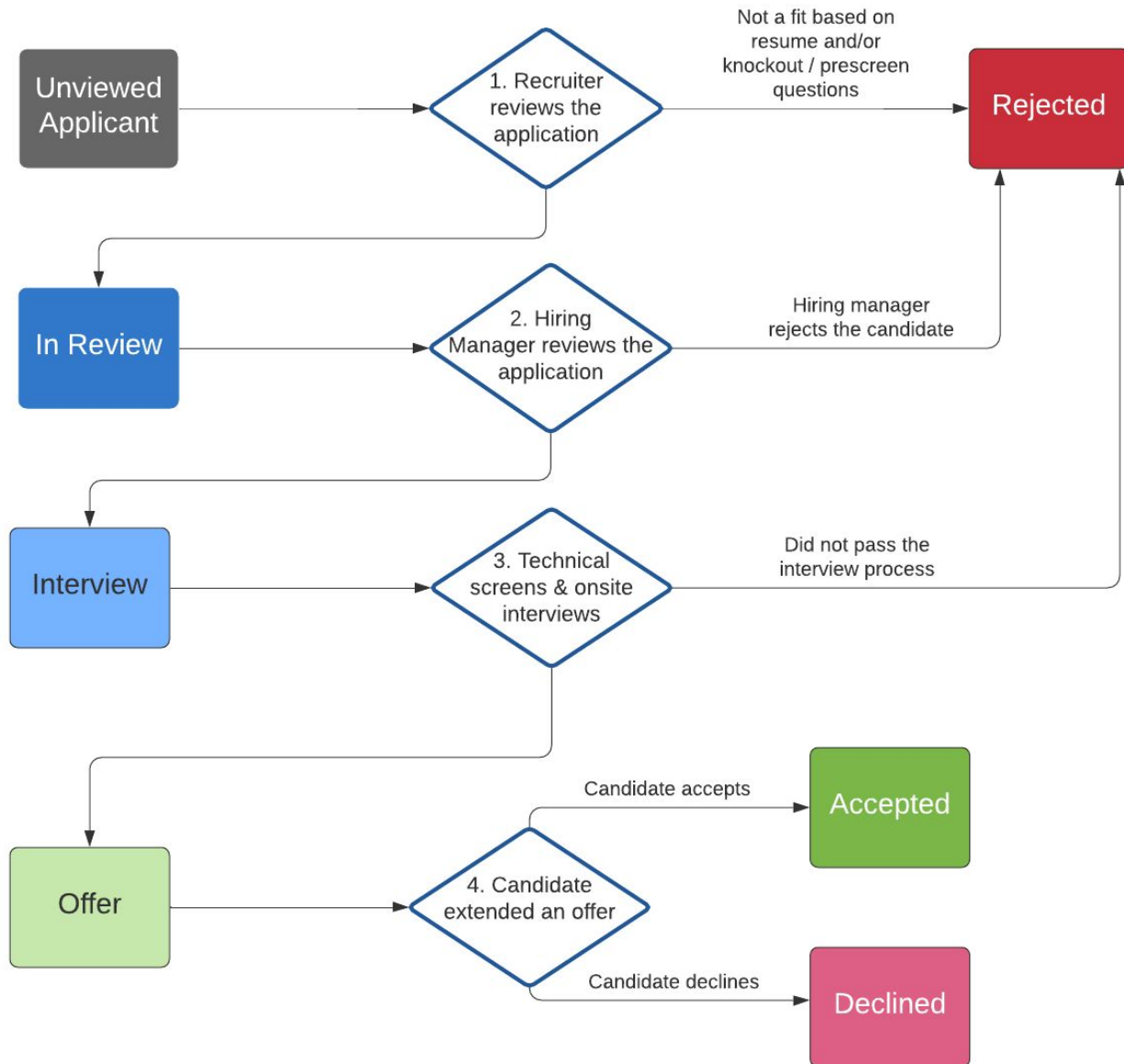
A typical Application Tracking System workflow. The ATS helps keep track of the status of each applicant. Very handy, when you have more than a few people applying!

Boxes are statuses that represent the status of the application. It starts as "Unviewed applicant", and will eventually become "Rejected", "Accepted" or "Declined". People - recruiters or hiring managers - move the status between one status, to another. For example, after a recruiter reviews the resume and the application, they will either move the status to "In Review" or "Rejected". The statuses and the workflows can be different for each company. A big focus in the functionality for ATSes is the flexibility of setting these up.

For a more detailed explanation on how an ATS system works, see the video The Truth About the ATS (it's not what you think) from Amy Miller. Amy is an experienced tech recruiter and the author of the

[Recruiting in Yoga Pants blog](#). She has recruited at Amazon, Google and Microsoft and shares the "behind the scenes" working of this process. In this video, she summarizes what an ATS is, and is not:

"An ATS is a system to keep me organized, as a recruiter. We need to actually know, at a given point, in their journey, and what I need to do next. For example, if I've got someone at the interview stage, have I set it up? At the offer stage, has the offer been approved yet? Without an ATS, I don't know what's going on. I don't know who is waiting on who, doing what thing. The ATS solves this problem.

The idea that the ATS is this mythical, genius, AI-infused tool is crazy. Anyone who has been in an ATS, and has used it for work is laughing at this idea. It's so ridiculous."

ATS Myths Busted

ATS systems are an excuse many resume builder sites and self-proclaimed career gurus use to promote false claims on how they work. These sites and people incentivise optimizing resumes in ways that might not make a difference.

Automated ATS rejections are one of the main fallacies many sites claim. Jobscan [writes](#): *"Whether that human recruiter ever sees your resume could depend on how well your resume is optimized for ATS algorithms.".* CNBC published the article [75% of resumes are never read by a human—here's how to make sure your resume beats the bots](#). This article only references sources from companies who make a living selling resume services and claim that they provide resumes to "beat" this system. There is no hiring manager or tech recruiter backing up the contents —good luck finding professionals giving their name to incorrect facts. Take this claim from the article:

"Most applications are removed from the equation because they are not formatted in a way these systems can read and interpret - a career expert with TopResume tells CNBC."

The claim that PDF resumes are "removed from the equation" is false, and the following advice, to use a Word document as a format, is also poor advice.
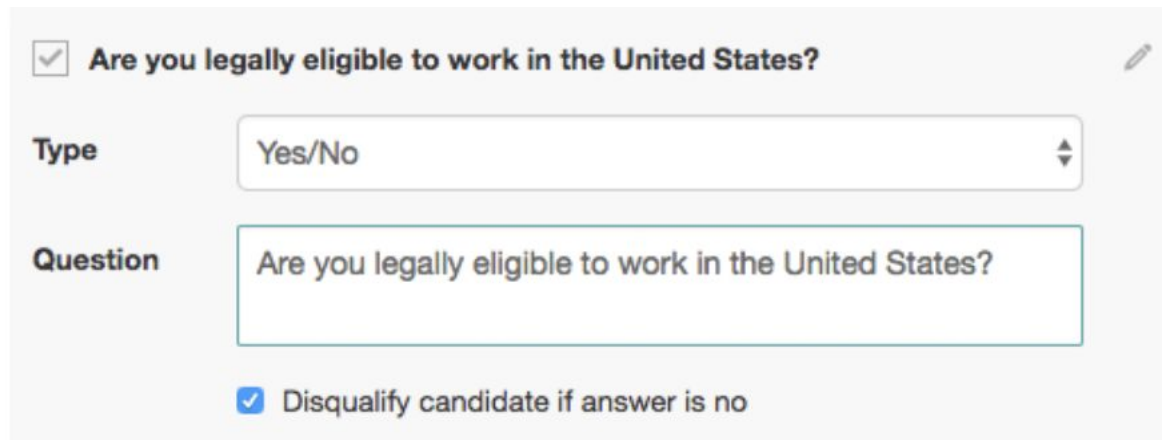
For the majority—if not all—of the tech companies, these claims are simply false. Recruiters *do* go through applications in the ATS interface, and will almost always scan resumes. The only time a recruiter will not look at your resume, is when there are enough other, qualified candidates and they stop processing resumes.

Knockout and prescreening questions are the one part, where there can be truth to the claim of "The ATS rejecting" a candidate. These are questions that recruiters set up and are intended to filter out people who are not eligible for the position, irrespective of their resume. For example, if a company does not sponsor visas, a recruiter might add a question to the application form saying, "Will you need sponsorship in order to work?" People needing visas will then be rejected, following the policy for the role. However, it was the recruiter who set up this rule, and the data needs to be explicitly provided by the candidate.

You can read more about how prescreening works by looking at the documentation for ATSes. Taleo calls this information gathering [prescreening questions](#), iCIMS names them [screening questions](#), Greenhouse refers to them as [custom questions](#), and they go by as [form application questions](#) with Workable. In all cases, recruiters can set up additional questions to gather information from applicants, and use this in the process. Here is the example of how a recruiter could set up a knockout question for

visa status, in Workable:

Setting up a knockout question in Workable. These are most frequently used to save time with candidates whom the company would not hire due to certain constraints.

For knockout and prescreen questions, you need to provide information on the job application form. This information is never taken from your resume.

ATS-compatible resumes are another fallacy. While it is true that most ATSes attempt to parse resumes, they don't use the contents to filter for anything during the screening process. Resume parsing is done to attempt to auto-fill candidate details. Resume parsing is also used to allow for boolean searching in the system later on outside the application process. However, beyond attempting to match personal details, and indexing keywords found in the resume, ATSes don't do anything clever. Both PDF and Word documents are parsed well enough. To get a sense of the type of searches a parsed resume supports, see this article on the iCIMS candidate search capabilities.

An "ATS-compatible" resume is more of a meaningless buzzword than anything else. Stating that a resume is ATS-compatible is like saying that a website is mobile phone compatible. Of course, it is. All websites are. And all resumes are ATS-compatible.

ATS resume optimization is a service several providers charge for - anywhere from $100 to up to $500 and beyond. They will claim that by matching the job description better, your resume will "bypass" the ATS algorithm, and a recruiter will look at your application. Claiming that there's a need to "bypass" the ATS is a false claim in the majority of cases for tech jobs. As of 2020, ATSes are still far more simple than even to understand what programming languages you've listed on your resume. Second, there is no "bypass". Your resume is in a queue, waiting for the recruiter to review.

The only popular ATS that does some type of rating based on parsed resumes is Taleo. Taleo can assign a requisition rank to your resume. This is a percentage number based on automated screening questions and your resume, and the job requisition overlapping. Recruiters can decide to sort resumes by this rank. However, this score is off for software developers. Few, if any, recruiters would rely on this to decide which resumes to look at, and which not to. None of the over a dozen tech recruiters and sourcers I've consulted have used such an automated rating to decide which applications to look at and which to pass on.

Tailoring your resume for the position is solid advice, though. This is the reason why you'd see results following ATS optimization techniques. If you tailor your resume for the job, you should see

better results. The ATS itself won't decide whether to move forward with your resume: the recruiter or hiring manager scanning your resume will. They will look to determine how relevant your experience is: and they'll do this in a few seconds. As the number of applications for roles is almost always high, tailoring your resume for the position and grabbing attention with the first scan, do make a difference.

Some tech recruiters predict more automated parsing and AI-assisted decision making to come for hiring. However, this is not here just yet. Even when it will be, the outcome shouldn't be different to a recruiter doing a quick resume scan, and marking the resume with a "Yes", "Maybe" or "No" outcome.

From the inside out: what recruiters say about ATS systems

I asked technical recruiters to share some common misconceptions about Applicant Tracking Systems and explain why those beliefs are incorrect. Here a few top ones they shared:

Filtering happens with human intervention, not automatically. *"The list of people in an ATS is organized by date when they applied. A real-life person then makes a decision. You can either be rejected: this can be due to not being a good fit based on your resume, or based on your answers to knockout questions. Or you progress in the process." (Amy Miller, author of the* [Recruiting in Yoga Pants](#) *blog, previously recruited at Amazon, Google and Microsoft)*

Tech recruiters would not trust an ATS to filter out candidates, as they could lose good people. *"No recruiter I know would trust an ATS with filtering, even the slightest. They don't want to risk losing otherwise perfectly good candidates. I've talked to many tech recruiters in the industry, and have never heard anyone relying on filtering based on automated parsing of CVs. The risk is too high." (Csudi Csudutov, founder of Mimox and tech recruiter for over 20 years)*

Submit PDFs CVs and you don't need to worry about resume filtering. *"ATSes are still so bad at filtering based on resume content that we get poorly formatted CVs, irrelevant profiles, or even blank pages. Personally, I suggest the best CV format being PDF. With Word documents, you risk ruining your format if the recruiter opens the document with another Word version or operating system. Agency recruiters also use Word CVs to remove your contact details to use it as business development material, so watch out!" (José Marchena, tech recruiter in London and host of the* [Coffee with a Recruiter](#) *podcast)*

Other sections in this chapter

This chapter sample ends here. In the book, it carries on with the following sections:

Referrals

This section is not available in the sample.

The priority of your resume

This section is not available in the sample.

Less competitive hiring pipelines

This section is not available in the sample.

Startup & small company hiring pipelines

This section is not available in the sample.

Mid-sized companies with HR support

This section is not available in the sample.

Job aggregators

This section is not available in the sample.

Recap

This section is not available in the sample.

Chapter 5: Resume Structure

There are two types of software developers: ones who have gotten their first full-time job, and ones who are looking to land that first one. Based on which group you are in, recruiters and hiring managers will care about different details.

For people without full-time experience, your internships, education details, projects, and achievements are what will set you apart from the many other applicants who are also looking to land their first jobs. As soon as you have that first job, your work experience and the professional skills you use day to day become far more relevant for recruiters and hiring managers.

Based on which group you are in, you'll want to structure your resume differently. And as you spend more time working professionally and start to build up more experience than can fit on a few pages, you'll have the pleasant problem of deciding which one of your past experiences *not* to talk about.

Structure for Interns, New Grads and Bootcamp Grads

This section is not available in the sample.

Structure with Work Experience

When you are no longer freshly out of school, follow this structure to make your resume easy to review.

- **Work Experience at or near the top of the page**. Your current title, company, and past few years of work experience is something the recruiter and hiring manager will want to glance at. Make it easy by adding it to the top, or close to it.
- **Have a Languages and Technologies section on the first page** that lists the relevant technologies. List things which you are an expert or, at the very least, proficient at. They could be domains, languages, or frameworks that the job description mentions. Don't bother listing non-relevant technologies, or listing your skill level.
- **If you have spent a long time at one workplace**, list out the key projects you shipped and the titles you've held there. Have you been promoted? Treat it as a new "sub-work" section, listing the projects you did at that point.
- **Education details become less important with seniority**. For education, slowly reduce the length of this section, as you have more work experience. With 1-3 years' experience, it's fine to have details on it, but with 5+ years, you'll likely just want to have your degree, date of graduation, and no more than one standout achievement, if it's still relevant. Summa cum laude can probably stay. GPA, courses, activities should all disappear.
- **Spend less space on old positions.** For people with 10+ years of experience, your work experience beyond 10 years is less interesting. What you did then is not representative of what you do now, and there's little point in listing obsolete technologies. Shorten these sections, and consider removing or skipping ones that are not relevant—especially if you were job-hopping a decade back. The resume should sell you, not show every place you ever worked at.
- **Extracurricular**. Add patents, publications, talks, standout open source projects, published projects, and other areas that could grab attention. In the case of open source and published projects, aim to be specific on why they are important. Close with hobbies and interests to make it personal. Keep the list of hobbies and interests short.

- **Certifications**. If you have certifications relevant to the job or the industry, list them below your work experience. Companies that work with governments that require certain professional certifications might place more focus on these areas. Also, be wary of the potential negative perception from listing a trivial-to-get certification—such as a LinkedIn programming language certification, which is a series of a dozen questions that can be repeated at any time.
- **Projects**: the more work experience you have, the less relevant outside-work projects tend to become. If you have something that really stands out, consider listing it under extracurricular, linking so that people reading the resume can inspect it. Use the results, impact, and your contribution format to explain why the project was relevant and impactful.
- **Interests**: depending on the length of your CV, you can add a few fun things to make your CV more "human". If you stick with a one-page format and you're short on space, you can skip this.

From the inside out: what recruiters typically look for in a resume

Victoria Farelly, who recruited for Uber, Booking.com, and ING explains how what recruiters typically look for in a resume is usually an extension of what the hiring manager asked them to screen for:

"A hiring manager will often say to you, as a recruiter: 'I want these five things, and if a person doesn't have these five things, I'm not hiring them.' If you're a good recruiter, you're there to advise them on the market and advise them that we have enough resources to take someone who only has three or four of those five things. Perhaps we have the resources to train or mentor them. Or perhaps they'll just pick it up in the first month.

When the hiring manager is more flexible on the "must-have things", you then look at if people have worked in similar environments, or on similar problems. For example, when hiring for Uber, you might look for signs that this person worked on something at scale. Did they work in multidisciplinary teams? What technologies have they been working with? And I'd look at not just your work projects, but also your personal ones. For example, if you've worked extensively with .NET at work, and knowing Java or Go is a must-have for the role, I'd expect to see some of those languages somewhere else, like in the projects or technologies section.

And I'd stress how what really makes you stand out is having a tailored CV for the position. *If you are applying for 20 different jobs, you should have 20 different CVs. Each one should be different and specific for that role. And while this might sound a "bad" thing to do, it's not. It's a necessary thing to do."*

For an actionable way to tailor your resume for a position, see the Keyword Check for That Position section within the Exercises to Polish your Resume section.

Languages and Technologies

"What languages and technologies is this person hands-on with?" This is one of the first questions recruiters and hiring managers have when they look at your resume. The easier it is to answer this question, looking at your resume, the better. There are a few common approaches in making this information clear— we'll look at three different ones.

Approach #1: Separate Languages and Technologies Section

The most common approach is listing relevant technologies for the position that you are proficient in a separate section. People tend to give this section various titles: Skills, Tech, Tools, and many others. The name is less important; the contents are more so.

By moving the languages and technologies you use to a separate section, you make it easy for the recruiter and hiring manager to verify what overlaps you have with the role. You shouldn't only list the technologies on the job advert, of course: but you shouldn't go overboard, either. Only list areas where you do have enough knowledge to do work day-to-day. I usually advise against listing the level of expertise, unless you have extremely deep knowledge of a relevant technology. I advise against using a points system as well. Also, avoid listing trivial technologies or ones that are niche, and have nothing to do with the job. Same goes with applications that are trivial to learn. As always, use good judgment.

Even when having a separate section to call out relevant languages and technologies, do mention key technologies in your work experience when you talk about specifics. This information will reinforce that you have had hands-on experience with a specific language or a given framework.

Before and after: languages and technologies

This resume is sent for a job advert for a full stack position. The job advert listed that knowledge of at least one OO language is a must, ideally between JavaScript, Go, or Java. Experience with a popular frontend framework, ideally, React.js, is an advantage, as well as having designed APIs. While not in the job description, the engineering blog describes how this company runs most of its infrastructure off AWS.

Before:

Relevant Skills

- Programming languages: Perl, C++, Java (expert), HTML5, CSS3, Bootstrap, JavaScript(ES6—proficient), React.js (expert), PHP
- Databases: MySQL (expert), Oracle SQL, MSSQL (proficient)
- Technologies: AWS, Bitbucket, GitHub, Visual Studio, MATLAB, Eclipse, Android Studio, phpMyAdmin, Adobe Photoshop, Oracle Fusion, Rational Rose, Sublime, Trello, Word

This section is a dump of all the technologies this person has touched in the past. Some of the listed ones include ones that are implicitly assumed—if you've used Java, you likely know how to use an IDE like Eclipse. And some technologies have no relevance: Rational Rose is a tool rarely used outside academia, and phpMyAdmin as a skill raises the question if you can manage PHP without a GUI interface. For Trello and Word: is there anyone who doesn't know how to use these?

The person is also using terms like "expert" and "proficient". This is a double-edged sword, as it implies that the person is not an expert in other languages. Also, talking with recruiters, the self-evaluation of people means little: several technical recruiters mentioned that people who rated themselves as an expert in a specific language would often get rejected based on not having enough depth, after being grilled in the depths of that language.

As a rule of thumb, avoid listing your expertise level. Instead, list only languages that you feel proficient with, and list your strongest languages and technologies first.

Improvement areas visualized:

Relevant Skills

- Programming languages: Perl, C++, Java (expert), HTML5, CSS3, Bootstrap, JavaScript(ES6—proficient), React.js (expert), PHP
- Databases: MySQL (expert), Oracle SQL, MSSQL (proficient)
- Technologies: Trello, Word, AWS, Bitbucket, GitHub, Visual Studio, MATLAB, Eclipse, Android Studio, phpMyAdmin, Adobe Photoshop, Oracle Fusion, Rational Rose, Sublime, Trello, Word (either trivial or overly specific technologies)

After:

Languages and Technologies

- Languages: JavaScript, Java, HTML/CSS, PHP, SQL
- Technologies: React.js, Bootstrap, AWS
- Other: API design, relational databases, unit, integration & E2E testing (adding skills relevant for the job listing)

The revised version is far cleaner. The formatting uses tabs, making it easier to scan. The tools and applications that anyone can pick up in a matter of hours are removed. The list is more relevant for the job description, and the languages that this person was actually hands-on with.

After talking with the person, it turned out that they have not used C++ or Perl in years, and they rated themselves as very rusty in these. Removing "old" languages makes sense both because they are not relevant. Also, languages that are no longer used can contribute to age bias - recruiters subconsciously assuming the person applying must either be old, or reluctant to pick up new languages.

Approach #2: work experience conveying languages and technologies

Another approach is to explicitly call out languages and technologies used at each of your positions, and not have a separate section for this. This approach helps convey the recency of the technologies you used, as opposed to having a big list of technologies—some of which you might not have used in a while.

This approach helps convey the recency of knowledge in a particular technology. Hiring managers and technical recruiters will have a better understanding of how up-to-date you likely are with certain stacks. This approach can work better when applying for tech companies hiring generalist software engineers, where it can be an advantage to show that you have moved between languages and stacks in the past.

Let's look at a snippet from a resume using this approach:

Software Engineer—Tax Returns

- Led development of two separate third party interface development projects for two of our biggest clients with Microsoft and EPAM, which were completed on time with 45% cost reduction.
- Automated error handling process of our restful API services that reduced one of our clients' support ticket rate by 80%.
- Technologies: Java, Swing, AWS, MSSQL

Software Engineer – Customer Happiness

- Created troubleshooting tools that improved the support team's efficiency by more than 30%.
- Designed and developed the data receiving module of the mass transaction uploader platform, which is used to pre-process and clean data before generating transaction files.
- Technologies: JavaScript, React, AWS, MongoDB

Weaving in the technologies into the descriptions is an approach that can also work well. It makes for a more natural reading experience. I would suggest to be consistent in where you mention the technologies, to make these easy to spot. Here is an example of this approach:

Software Engineer—Tax Returns

- Led development of two separate third party interface development projects for two of our biggest clients with Microsoft and EPAM, which were completed on time with 45% cost reduction. I build the system using Java and Swing.
- Automated error handling process of our restful API services that reduced one of our clients' support ticket rate by 80%. Used Java and MSSQL, deploying on an AWS stack.

Software Engineer – Customer Happiness

- Created troubleshooting tools that improved the support team's efficiency by more than 30%. Built the tools with React and MongoDB.
- Designed and developed the data receiving module of the mass transaction uploader platform, which is used to pre-process and clean data before generating transaction files. Used JavaScript / Node.js, deploying on AWS.

Approach #3: splitting out not-so-hands-on languages and technologies

The downside of having a list of languages and technologies is that it does not differentiate between ones that you are hands-on with, and ones where you would need a refresher. In this case, adding technologies that you are a bit more rusty with—but differentiating these—can be an option.

Here's an example of this, where a person is applying for a position for a company that is heavy on Ruby. They've done this in the past and wouldn't mind picking it up again, but their Ruby knowledge is

not on the same level as JavaScript and Java, which they both use day-to-day.

Languages and Technologies

- Languages: JavaScript, Java, HTML/CSS, SQL
- Technologies: React.js, Bootstrap, AWS
- Working knowledge of: Ruby, Rails, PHP

Tell a Story

This section is not available in the sample.

The Summary Section

This section is not available in the sample.

Structure for Senior and Above People

This section is not available in the sample.

Recap: Actions to Improve Your Resume

This section is not available in the sample.

Table of Contents for The Tech Resume Inside Out

Sample for The Tech Resume Inside Out. Get your copy at TheTechResume.com.