



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

Sold to  
wchen1222@gmail.com 

---

Copyright © 2020 by Gergely Orosz

All rights reserved. No part of this book may be reproduced or used in any manner without written permission of the copyright owner except for the use of quotations in a book review. For more information, please address: [book@thetechresume.com](mailto:book@thetechresume.com).

ISBN: 978-1-63649-416-6 (ebook)

FIRST EDITION, v1.0

[www.TheTechResume.com](http://www.TheTechResume.com)

ISBN 978-163649416-6



A standard 1D barcode representing the ISBN 978-163649416-6. The barcode is composed of vertical black lines of varying widths on a white background. Below the barcode, the numbers "781636" and "494166" are printed, likely as a check or reprint code.

---

# Contents

<b>Acknowledgements</b>	<b>6</b>
<b>Introduction</b>	<b>7</b>
Who This Book is For	7
What This Book is Not	7
How to Read This Book	9
<b>PART 1: RESUMES AND THE HIRING PROCESS</b>	<b>10</b>
<b>Chapter 1: Why Resumes and CVs are Important</b>	<b>11</b>
The Goal of a Resume	11
Good Resumes, Great Resumes	13
Why LinkedIn is Not Enough	15
A Resume Is (Still) Not Enough	16
<b>Chapter 2: The Hiring Pipeline</b>	<b>17</b>
People in the Recruitment Process	17
The Typical Hiring Pipeline	20
The Applicant Tracking System	22
ATS Myths Busted	24
Referrals	27
The Priority of Your Resume	31
Less Competitive Hiring Pipelines	34
Job Aggregators	38
Recap	39
<b>Chapter 3: Resumes and COVID-19</b>	<b>40</b>
The COVID-19 Job Market	40
The COVID-19 Hiring Pipeline: What Hiring Managers Say	41
COVID-19 and New Grads/Interns	42
Experienced Engineers and The Need For Resumes During COVID-19	42
<b>PART 2: WRITING THE RESUME</b>	<b>43</b>
<b>Chapter 4: Tech Resume Basics</b>	<b>44</b>
The First Glance	44
Ground Rules	46
Simplicity and Consistency	47
Avoiding Biases: Personal Details and Photos	48
Recap: Actions to Improve Your Resume	51
<b>Chapter 5: Resume Structure</b>	<b>52</b>
Structure for Interns, New Grads and Bootcamp Grads	53
Structure with Work Experience	56
Languages and Technologies	58
Tell a Story	61
The Summary Section	63
Structure for Senior and Above People	65
Recap: Actions to Improve Your Resume	69

---

<b>Chapter 6: Standing Out</b>	<b>70</b>
Results, Impact and Your Contribution	71
Don't Be Humble	75
Write a Resume for That Job	79
Different Companies, Different Focus	82
Keyword Stuffing	86
Recap: Actions to Improve Your Resume	88
<b>Chapter 7: Common Mistakes</b>	<b>89</b>
Poor Format	89
Forgetting About Your Audience	92
Unnecessary Details	93
Links	97
Recap: Actions to Improve Your Resume	101
<b>Chapter 8: Different Experience Levels, Different Career Paths</b>	<b>102</b>
Current college and university students	102
Bootcamp grads	103
Career Changers	107
Career Breaks	110
Senior and Above Engineers	112
Tech Leads	113
Engineering Managers	114
<b>Chapter 9: Exercises to Polish Your Resume</b>	<b>116</b>
Write Two Different Resumes	116
Find Out the Impact of Your Past Projects	116
Do a Grammar Check, Not Just a Spellcheck	117
Ask for Friends or Family to Proofread	117
Get Feedback on the Internet	118
Keyword Check for That Position	119
Recap: Actions to Improve Your Resume	120
<b>Chapter 10: Beyond the Resume</b>	<b>121</b>
LinkedIn Profile	121
GitHub	126
Technical Blogs	129
StackOverflow, Twitter, Instagram, Quora and Other Social Sites	129
Cover Letters	130
Recap: Actions to Improve Your Application Beyond the Resume	134
<b>Part 3: EXAMPLES AND INSPIRATION</b>	<b>135</b>
<b>Chapter 11: Good Resume Template Principles</b>	<b>136</b>
Good Resume Template Layout and Principles	138
The Top-Down Layout	139
Two-Column Layouts	141
How Recruiters and Hiring Managers Scan Resumes	144
The Results of Using a Good Resume Template	149
Recap: Good Resume Template Principles	152

---

---

<b>Chapter 12: Resume Templates</b>	<b>153</b>
Resume Generators vs Resume Templates	153
Standard Resume: By Developers, for Developers	157
Template Reviews	159
Recommended Resume Templates	159
Other Resume Templates	167
<b>Chapter 13: Resume Improvement Examples</b>	<b>176</b>
Software Engineer with 2 Years Experience	177
Machine Learning Engineer with 5 Years Experience	183
Backend Developer with 6 Years Experience	190
Remote Software Engineer With 8 Years Experience	195
SRE Engineer with 20 Years Experience	202
Other Real Resume Examples	208
<b>Chapter 14: Advice for Hiring Managers on Running a Good Screening Process</b>	<b>209</b>
Know that your screening process is broken	209
Treat recruitment as a partner function	211
Write a good job description	212
Have a feedback loop	213
Learn how others do it	214
<b>Further Reading</b>	<b>215</b>
<b>Conclusion</b>	<b>217</b>

---

## Acknowledgements

This book has been written with significant input from some of the best technical recruiters, engineering managers and engineers within my network—many of them considered industry veterans. They were all generous with their time and feedback and made the book significantly better.

On the recruiting side, thank you to Amy Miller, Angelica Shah, Blake Stockman, Csudi Csudutov, Chukwuemeka Ugorji, Claire Taylor, José Marchena, Jorick Thijs Polderman, Keki Mwaba, Ken Liu, Konstanty Sliwowski, Sebastian Prieto Tovar, Sofi Djelassi, Victoria Farrelly, Veronika Nagy, and Yinka Coker. All of you have scanned more developer resumes than myself, and your insights and feedback helped make the book more accurate and helpful.

I am grateful for the feedback from engineering managers and engineers James Stanier, Jordan Bonnet, Manan Patel, Matt Dickenson, Monica Lent, Neville Kuyt, Pat Kua, Randall Kanna, Rodrigo Pimentel, Stuart Davidson, and Steve Ball. Thank you for both the reviews, the corrections and for confirming where you also would give similar advice.

The feedback and encouragement from the early beta book readers has been great. People who have helped with feedback and suggestions include Aleph Melo, Anmol Singh Jaggi, Christopher Paika, David Kiss, Fabien Townsend, Jamie Robertson, Maroof Haque, Munasir Priom, Oliver Mensah, Patrick Doyle, Simon Van de Water and Waseef Akhtar. Thank you! I made several improvements in the book—such as adding Recap sections at the end of many chapters—after these reader suggestions.

I'd like to give special thanks to the book's editor, Mike Reeves-McMillan, who worked with me from the other side of the world to create something that people will find pleasant to read.

# Introduction

There's a myth of it being easy to get a job as a developer. While there are times where this might be true—especially if you are a senior or staff software engineer working at a well-known tech company—this is not the case for most developers. As soon as you get out on the job market and start directly applying for jobs, the feeling of having it easy quickly disappears.

This book will help you craft a developer resume that represents you fairly, plays to your strengths, and increases your chances of getting to that recruiter call.

## Who This Book is For

This book is tailored for people applying for developer jobs with tech companies in the US, Europe, and India, for companies where the language used for business is English. The book is especially relevant for companies with US headquarters: well-known tech companies like Google, Amazon, Facebook, Apple, Netflix, Google (also referred to as FAANG<sup>1</sup>), unicorns like Stripe, Uber, Airbnb, Dropbox and fast-growing tech startups. The principles of the guide likely translate to other companies. However, different industries and different companies might follow different processes, and look for different things in a resume.

The book is written for software developers, tech leads and engineering managers. Most of the content is tailored from new grads, all the way to experienced software developers, with dedicated sections for leads and engineering managers. The book assumes that you already have the relevant experience for the positions you are applying for.

Everything presented in this guide is opinions that might or might not apply to the country and industry you are applying in, or the company you are applying to. Make sure to do your own research and get feedback from your network to tailor your resume or CV. There is no one-size-fits-all approach.

## What This Book is Not

This book is not a definite guide on what works, and what doesn't, when applying for any specific company. Though the author and contributing authors have worked at a variety of tech companies, this does not make it an official guide for any of the companies mentioned in the book.

This goes back to how hiring works. As we'll cover in [Chapter 2, The Hiring Pipeline](#), hiring is a process that can differ not only per company, but also per hiring manager. Also, the likelihood of you receiving a follow-up call after applying for a position is dependent on many other factors that are impossible to predict. These include how many other qualified applications there are, what the timeline to fill the position is, how many open positions there are, and many others.

Software engineers realize that there are no one-size-fits-all solutions on what language to use to build a website, or what architecture to follow when designing a complex application. In a similar way,

---

<sup>1</sup> FAANG: Facebook, Apple, Amazon, Netflix, Google. FAANG companies are known for the impressive stock growth they have shown in recent years. Engineers working at these companies receive hefty equity compensation on top of base salaries, making them some of the most attractive companies to work at—and the most competitive to get into.

---

there is no one-size-fits-all resume template that will yield the best results, or resume contents that will always result in a recruiter call.

The book will not be able to answer why you don't hear back from companies you apply to. However, it does give context on how the process works, and advice that should help you navigate the resume writing process better, and with more structure and confidence.

## How to Read This Book

The book is split into three different parts. I recommend skimming the book once, choosing one or two resume templates, then writing or re-writing your resume, as you go through the exercises laid out in Part 2: Writing the Resume.

**Part 1: Resumes and the Hiring Process** shares context on why resumes are important, and how the hiring process works at most companies. If you have not been a hiring manager yourself, I recommend starting with this section. In order to deliberately write a good resume that recruiters and hiring managers will notice, knowing how the process works and who you are writing the resume for is important.

**Part 2: Writing the Resume** presents actionable advice on how to write a good resume. Each of the chapters in this part focus on a specific area within your resume. This part is heavy on examples.

**Before and after examples** showcase real-world, but anonymized resume snippets as their author originally wrote them. Improvement areas are visualized with highlights, and the improvements afterwards are also made clear.

**Recap: actions to improve your resume** sections close each of the chapters in Part 2. These are practical checks and exercises you can do to make improvements to your resume.

**Part 3: examples and inspiration** provides templates to get started with writing your resume, as well as anonymized developer resume examples to kick start you with writing your own resume.

### From the inside out: insights from people running the hiring process

The book contains a dozen “from the inside out” sections. These are the thoughts and perspectives of hiring managers, tech recruiters and other people involved in the recruitment process on how and why things are done. These sections offer additional and interesting perspectives.

---

## PART 1: RESUMES AND THE HIRING PROCESS

Before jumping into the specifics on how to write a good developer resume, we need to take a pause. Why do you need to write a resume? What does a good developer resume look like, anyway? And why not just use your LinkedIn profile?

Once you submit a resume, what will happen to it? Will a recruiter look at it? Or a hiring manager? Or someone else—perhaps an automated process making a decision?

This part shares the context that recruiters and hiring managers would call trivial. However, if you are not a technical recruiter or a hiring manager, much of the details in this part will be eye-opening. Like what hiring managers see as good resumes, or what happens to your resume behind the scene.

In [\*\*Chapter 1: Why Resumes and CVs are Important\*\*](#), we discuss what the goal of a resume really is. Spoiler it's not to showcase all of your past experience. We cover what a good resume is like, and why LinkedIn might not be sufficient to get a job when you directly apply.

In [\*\*Chapter 2: The Hiring Pipeline\*\*](#), we share the bigger picture of how the interview process looks—so you can better understand why resumes are important.

In [\*\*Chapter 3: Resumes and COVID-19\*\*](#), we reflect how COVID-19 has changed the job market, and what this means from new grads to experienced engineers. This chapter of the book is very much specific to the 2020 COVID-situation and as the world gets back to a new normal, the contents of this section might become less relevant.

Let's start by diving into why resumes are important at all, at times when LinkedIn is such a popular way to represent yourself professionally.

# Chapter 1: Why Resumes and CVs are Important

I have reviewed over a thousand resumes the past few years—or, as they call them in Europe, Asia and several other places, CVs<sup>2</sup>. As a hiring manager, I glanced through most of these, taking 10-20 seconds to decide on whether to read more in-depth. In a minute or two, I always decided whether to proceed with a call with the applicant. For contrast, the average recruiter spends around 7 seconds<sup>3</sup> to scan a resume and decide if it's a reject, or if they'll read further.

For a week in May 2020, I carefully reviewed more than 300 resumes, spending far more time on each than a recruiter or hiring manager would. These resumes ranged from senior engineers to new grads. The people writing these had put time and effort into building what they thought was a good resume. Still, I had plenty of feedback on each.

**The majority of developer resumes I have seen did a poor job representing the person behind the profile.** As a hiring manager in a hurry, with more than a hundred resumes to go through, I would have passed on many of the otherwise strong and qualified candidates. This was because their potential did not come across on the resume. Recruiters, who often are less technical, would have passed on even more of them.

You write a resume when you are *proactively* applying for a company: when you are actively searching for a job. And when you apply directly, you want a resume that represents you well, on the first glance. If you don't put in the time and effort, don't be surprised if you'll be passed over for someone who is just as qualified, but has a resume that sells them better for that specific position.

## The Goal of a Resume

Why do we spend so much time crafting a resume? What is the goal of this exercise? Many people just create a resume, as this is what's needed to apply to a job. You need to upload an attachment in the application form. So several people google "resume template", fill it out with their experience, and submit. They do the work but don't think about the goal.

**The goal of a resume is to get you to an interview for that position** with a recruiter or hiring manager from a specific company. That's it—nothing more, nothing less. It's to get your foot in the door. It's not a goal to share your complete work history or represent everything you did as accurately as possible.

The goal of the resume is to sell you for that specific position—and do this in the less than 10 seconds that the recruiter or hiring manager will be reading it. Keep this purpose in mind, as all the content in this book will revolve around this goal.

### From the inside out: how hiring managers typically go through resumes

Nevile Kuyt is a hiring manager with two decades of hiring engineers behind him. He has reviewed thousands of resumes and shares his process on screening inbound resumes:

*"I like to use the "yes/maybe/no" model for this process. As a hiring manager, I go through a pile of*

<sup>2</sup> CV stands for Curriculum Vitae. In Latin, this means "course of life".

<sup>3</sup> Source: HR Drive [Eye tracking study shows recruiters look at resumes for 7 seconds](#)

resumes, and my goal is not to find the unique snowflake who would be perfect—I don't have time for that. Instead, I want to get that pile of 300 resumes down to the handful I'll interview, and do that as quickly as possible because I have other stuff to do.

I create three piles:

1. **Yes** pile: these resumes jump out at me as being a very good fit. I'll interview them. No more than 5 in this pile, and if I find a 6th, the weakest moves to the:
2. **Maybe** pile: not a perfect fit, but close. If I don't get enough "Yes" ones, I'll go through the maybe pile. No more than 5 in this pile. Everything else goes into the:
3. **No** pile: I'm really looking for a reason to put the resume in the NO pile. Poor spelling, grammar, no relevant experience in the last two years, different tech stack? If the "Yes" pile is big enough, those will get the resume onto the "No" pile."

**The goal of your resume should be to "jump out" as a Yes pile candidate.** The Yes/Maybe/No approach is typical of how hiring managers and recruiters screen resumes. Assuming you are a fit for what the job is looking for, you'll want to tailor your resume to maximize the chances of this match being clear on the first scan. With many applicants and a poorly written resume, there's a good chance yours will end up in the "Maybe" or the "No" pile. And if there are enough "Yes" candidates who pass the phone screen, you might not get that next call, even if you are on the top of the "Maybe" pile.

## Good Resumes, Great Resumes

**Good resumes** are ones that represent you in the fairest possible way, maximizing your chances of getting through the resume screening process—assuming you are qualified. However, even with a good resume, you won’t win over a recruiter every time. This could be for a variety of reasons, the main ones being that your profile doesn’t match what the company is looking for, or there are stronger profiles competing with yours. However, with a good tech resume, you won’t miss opportunities due to not representing yourself well enough.

**Great resumes** are ones where you grab the attention of the recruiter or the hiring manager, and they immediately want to talk to you. Unfortunately, this has little to do with the format, and all to do with the content. Great tech resumes are ones that showcase rare and valuable experience or skills that are *exactly* what the job needs. If you happen to be an engineer who is an expert in a niche framework, and you apply directly to a small company who sees few applications, and who are rebuilding their product using this same framework, the recruiter won’t believe their luck.

Great resumes are not generic: they are great for *that* specific role. While it’s tempting to think that having worked at Google makes your resume great, this won’t be the case for all positions. Going back to the previous example: if you’ve worked at a senior frontend developer at Google, your resume won’t necessarily count as great for a backend position that is looking for deep expertise building developer tooling.

Great tech resumes are built on the years of specific experience you gather that make you stand out from the crowd of software engineers, industry-wide. This guide will not help you with that. However, it will help you craft a good tech resume. A resume that will increase your chances of getting your foot in the door of a company, where you can level up, before moving to the next one. All the way until you have the skills and experience for a great tech resume.

Great resumes can also be ones that show solid career progression towards the current role. There’s a trail of promotions, taking on more responsibility or taking on more initiative, throughout the years. They are a stark contrast to journeyman resumes.

**Journeyman resumes** tell a story with zero, or close to zero progression. For long stretches of time, little to no career progression is visible. For high-growth companies, journeyman resumes can be a reason for rejection, as recruiters and hiring managers look for signs of progression. These signs could be some level of professional growth, as well as taking on more responsibilities or solving complex problems.

If you have a resume that reads like a journeyman resume, think about why this is. Are you not telling your story right? Or have you actually not been growing in any meaningful way for the past few years? It’s unlikely that you have not grown at all, so visualize the increase in responsibility, any vertical moves you have made, and the times when you have been promoted.

## **From the inside out: does having worked at Google make my resume great?**

Out of the hundreds of developer resumes I reviewed, a particular one stood out. This resume was from someone who spent a combined five years at Apple and Google before moving over to a small company. And yet, they reached out to me, as they were not getting many interviews and couldn't find out why.

True, the resume caught my eye, just like every recruiter's eye, so I read it in detail. But with every paragraph, my initial excitement faded. The resume painted the picture of someone who did mediocre work at the current small company. They described how they were working on a hard problem and built a solution end-to-end. They then linked to a barely-working web prototype, which looked and felt amateur when I tried it out. Reading further, there was very little mention of the specifics of the work at Google or Apple. And even what was written was making it seem like they only played a side role on internal projects at Google and Apple.

This person might have still gotten interviews, had they not been interviewing in the Valley. Here, Google and Apple are well-regarded, but the name itself won't always get you through the door. The resume also needs to sell you. This resume did no selling—it only did underselling.

When I talked with this person, they explained that they were a backend engineer, and frontend development was something they rarely did—and did not enjoy. They explained all the complex challenges they had solved with the small company on the backend. They implemented better text recognition algorithms, text clustering and natural language processing solutions. They did remarkable research, built a solid text analysis system—but mentioned nothing about this in their resume. Instead, the reader saw the result of their frontend work—which was, indeed, the work of a few weeks, but reflected very little of their real strength in backend development.

A similar story emerged when we talked about what they worked on at Google and Apple. They did build key components—one of them was even open sourced—but they did not mention this on the resume.

This person reworked their resume by identifying their most important personal contributions and achievements. The new resume looked far stronger: talking about how they built a complex backend system at the small company, how they were tech lead of a team of three on an internal project at Google, and how they solved a difficult and impactful technical challenge at Apple.

**Structuring your past experiences to showcase your best personal contributions is key in getting noticed**—even when you have strong companies on your resume. Hiring managers are looking for people who can help them with their current challenges of building software and shipping value.

The updated resume was what I'd call a Great Tech Resume, because it represented the person fairly, and showcased their impactful contributions at well-known companies well. After this, the interviews kept coming in. While it's not exactly hard to achieve this success for someone who has worked at Google, it's a reminder that presentation and storytelling make a big difference.

## Why LinkedIn is Not Enough

Several developers I talked with have no resumes. They have a LinkedIn profile that has served them well until now. When applying for a job, they just export this profile as PDF and send it off. While this approach will work in the cases where there are few applicants, or you have standout experience, it won't serve you well in all cases.

**LinkedIn is a tool for getting sourced when you are in demand: it is a poor tool for directly applying.** It has poor use of space, does not optimize for recruiters to scan it. And frankly, it shows you're not *really* interested in that particular job you applied for.

There are other downsides when using LinkedIn as your resume:

- **Not able to tailor to a specific position.** In many cases, your background and experience make you qualified for multiple positions. For example, a developer with fullstack experience can decide to apply to frontend, backend, or fullstack roles. For each of these roles, you'll want a tailored resume. With LinkedIn, you are stuck with a generic one that will not do as well as if you customized it.
- **Lack of history** or versioning. With resumes, you can have resume versions stored that you iterate on. With LinkedIn, there is no versioning.
- **Overly verbose** generalization. The page count is high for resumes generated with LinkedIn. The design wastes a lot of space.
- **Lock-in** with LinkedIn. There is no guarantee that the LinkedIn resume generation feature will be there in the future, or that it will work as it does today. When using a text editor, you have less lock-in. When using a resume generator, you have lock-in, but can reasonably expect the functionality to work in the future as well.

To be clear: LinkedIn resumes work at times when it's a developers' job market. They also work when companies struggle to get anyone interested in applying. Heck, in those times, you can skip this guide, put a text file together with your name, email, years of experience, and you could easily get invited for a technical interview. That's how things went in the late '90s, before the Dotcom bust. To some extent, the same happened around early to mid-2010s, when huge amounts of VC money poured into many startups, who had to hire fast. If you're in this kind of a market, you need to worry far less about resume formatting. If not—that is where this book can help you.

If you want to get a job via a direct application, build that resume and customize it for *that* job. And do keep your LinkedIn up to date: see the [LinkedIn Profile section](#) for more advice.

## A Resume Is (Still) Not Enough

I have gotten several messages, where people think that the format of their resume is the only thing that is getting in the way of getting that recruiter call. Messages such as this one:

*"I have been applying to Facebook in London, Stripe in New York, and Snap in LA. I haven't heard back from any of them. Can you help me fix my resume so I get an interview?"*

**A well-written resume is required, but it is not sufficient** to get a recruiter call. There are several other factors that are outside your control. These include:

- **If the position is still open.** The position you submit your application to might have already been filled—without you knowing about it.
- **Your competition.** You might think you have a strong profile, but what if there are dozens of people with more experience, with stronger skills applying for the same position? You won't get an interview. Also, for remote-friendly and remote-only jobs, the competition will be far more intense.
- **Being local vs needing relocation.** If a position sees hundreds of applications, the local candidates might get preference. They won't need relocation, which means they could start faster. Also, the lack of relocation makes this cheaper for the company.
- **Your visa status.** When a company has to apply for a visa for you to work in the country, this takes time and money. Companies usually only sponsor visas for senior positions, where they cannot find candidates without needing visas. Even many of the big tech companies only sponsor visas for senior or above positions.
- **Your seniority.** Certain positions only look for people with certain seniority—even if they do not advertise this externally. You might be considered not having enough experience for a position. This is frequently the case when you are a new grad, or have less than two years of experience, applying for positions that do not consider people without a few years of industry work behind them. In more rare cases, the recruiter might decide you seem to be overqualified for the position, with all your experience.
- **Luck.** Blind chance also plays a role, especially when your resume competes with hundreds of others. There are ways to make luck less of a factor—see [the section on referrals](#) (TODO), which can make the biggest difference in eliminating this factor and getting to a recruiter screen.

This book aims to help you craft a resume that represents you fairly. While following the advice will hopefully help you see higher response rates, there is no “magic formula” for getting that next call. At the same time, the more you know about what is happening behind the scenes, the more you can write your resume for the people who will be reading them. To understand who the key people are, let's dive into how the hiring process typically works at tech companies, and who might be reading your resume.

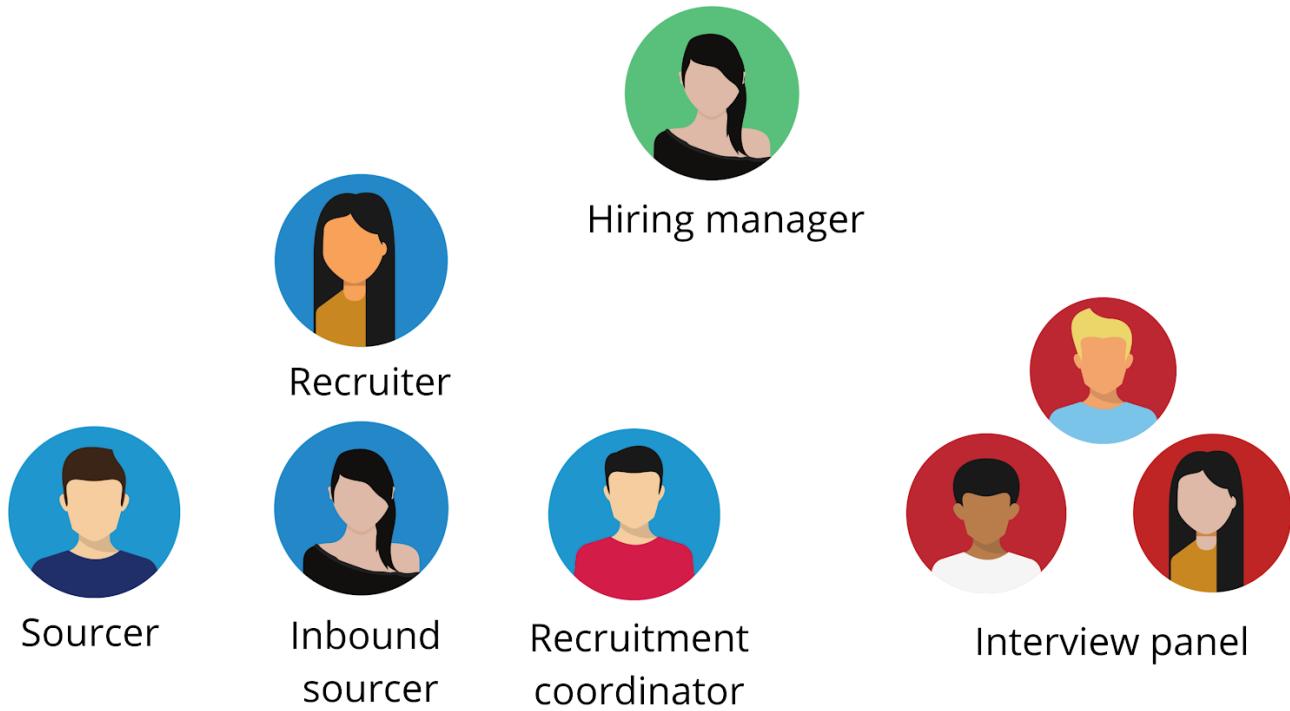
## Chapter 2: The Hiring Pipeline

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.



- **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.*

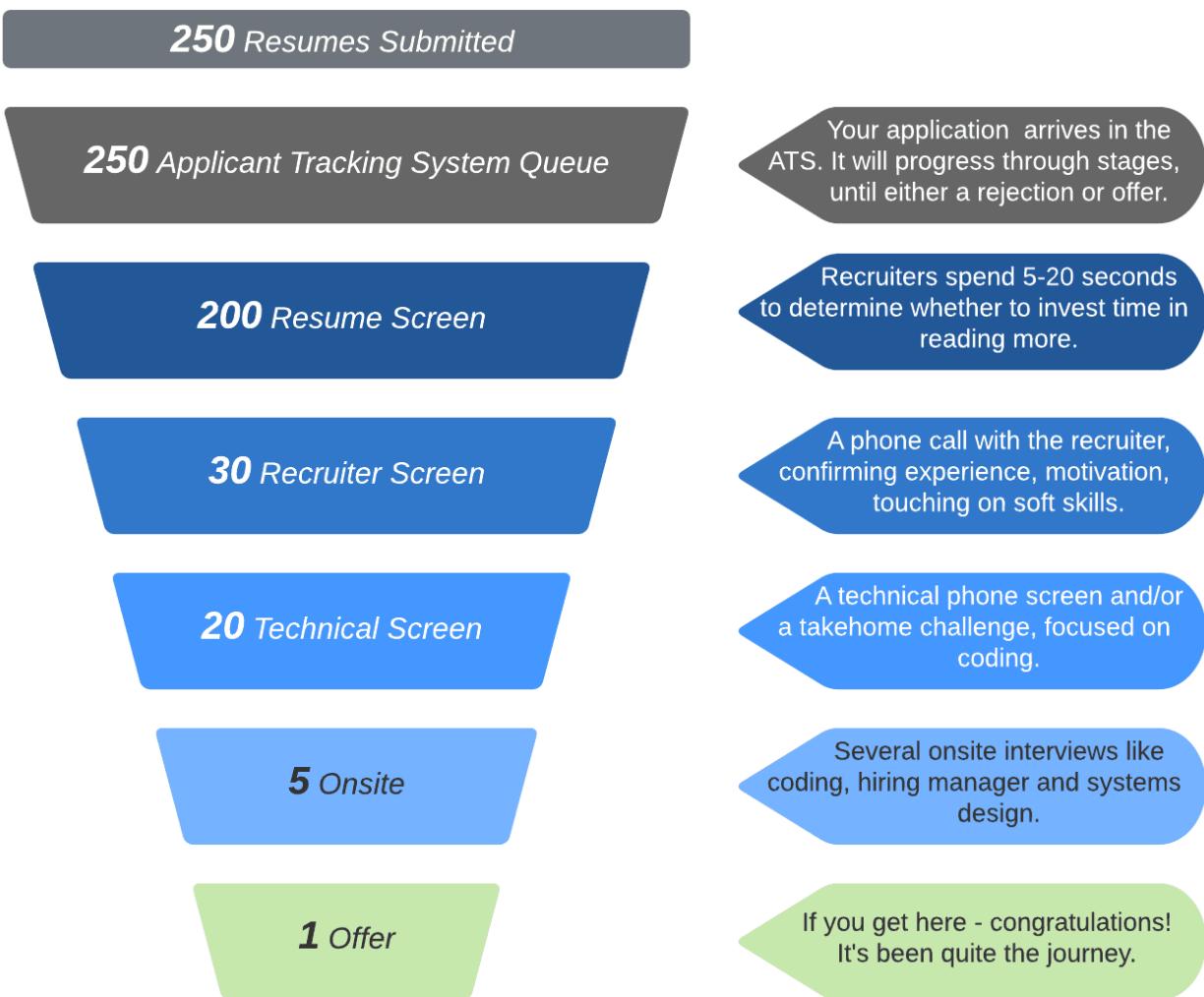
*Whereas when I was at Uber, I was working with hiring managers directly on actually crafting what that recruitment process was, specifying the outputs of interviews. In this case, I was able to tell candidates what to expect at the level of focus areas—without giving away the exact questions themselves. 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:



*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 is 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.

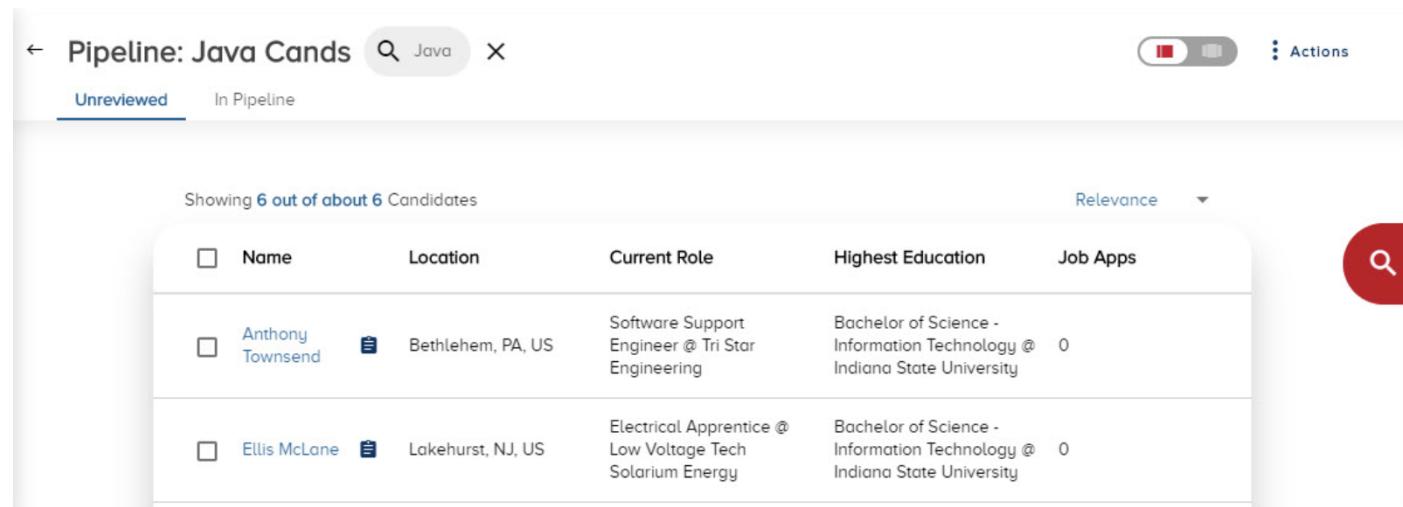
## The Applicant Tracking System

Applicant Tracking Systems (ATSSes) are a tool many candidates speculate about, and ask the question: do ATSSes 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:

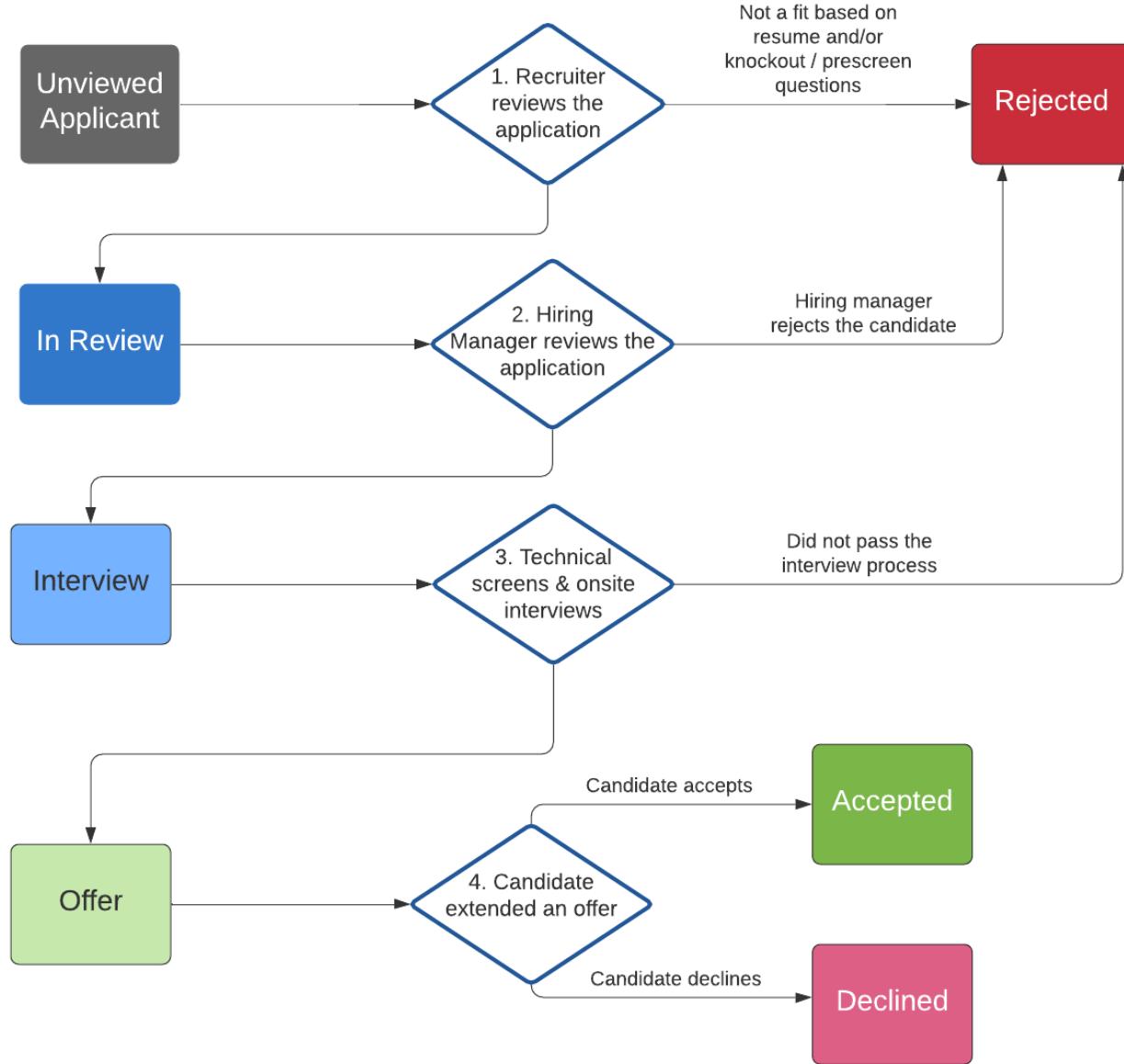


A screenshot of the iCIMS Pipeline view. The top navigation bar shows 'Pipeline: Java Cands' with a search field containing 'Java' and an 'Actions' button. Below the navigation, there are two tabs: 'Unreviewed' (which is selected) and 'In Pipeline'. The main area displays a table titled 'Showing 6 out of about 6 Candidates'. The table has columns for Name, Location, Current Role, Highest Education, and Job Apps. A red magnifying glass icon is positioned to the right of the table. The data in the table is as follows:

	Name	Location	Current Role	Highest Education	Job Apps
<input type="checkbox"/>	Anthony Townsend	Bethlehem, PA, US	Software Support Engineer @ Tri Star Engineering	Bachelor of Science - Information Technology @ Indiana State University	0
<input type="checkbox"/>	Ellis McLane	Lakehurst, NJ, US	Electrical Apprentice @ Low Voltage Tech Solarium Energy	Bachelor of Science - Information Technology @ Indiana State University	0

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:

**Are you legally eligible to work in the United States?** 

Type	Yes/No
Question	Are you legally eligible to work in the United States?

Disqualify candidate if answer is no

*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)

# Referrals

If you are surprised at how difficult getting to a recruiter screen is, I have good news, and I have bad news. The good news is that referrals make this much easier—if you have one. The bad news is that without a referral, it's actually more difficult to even have a human look at your resume or to get a recruiter call.

## **Referrals, references and background checks**

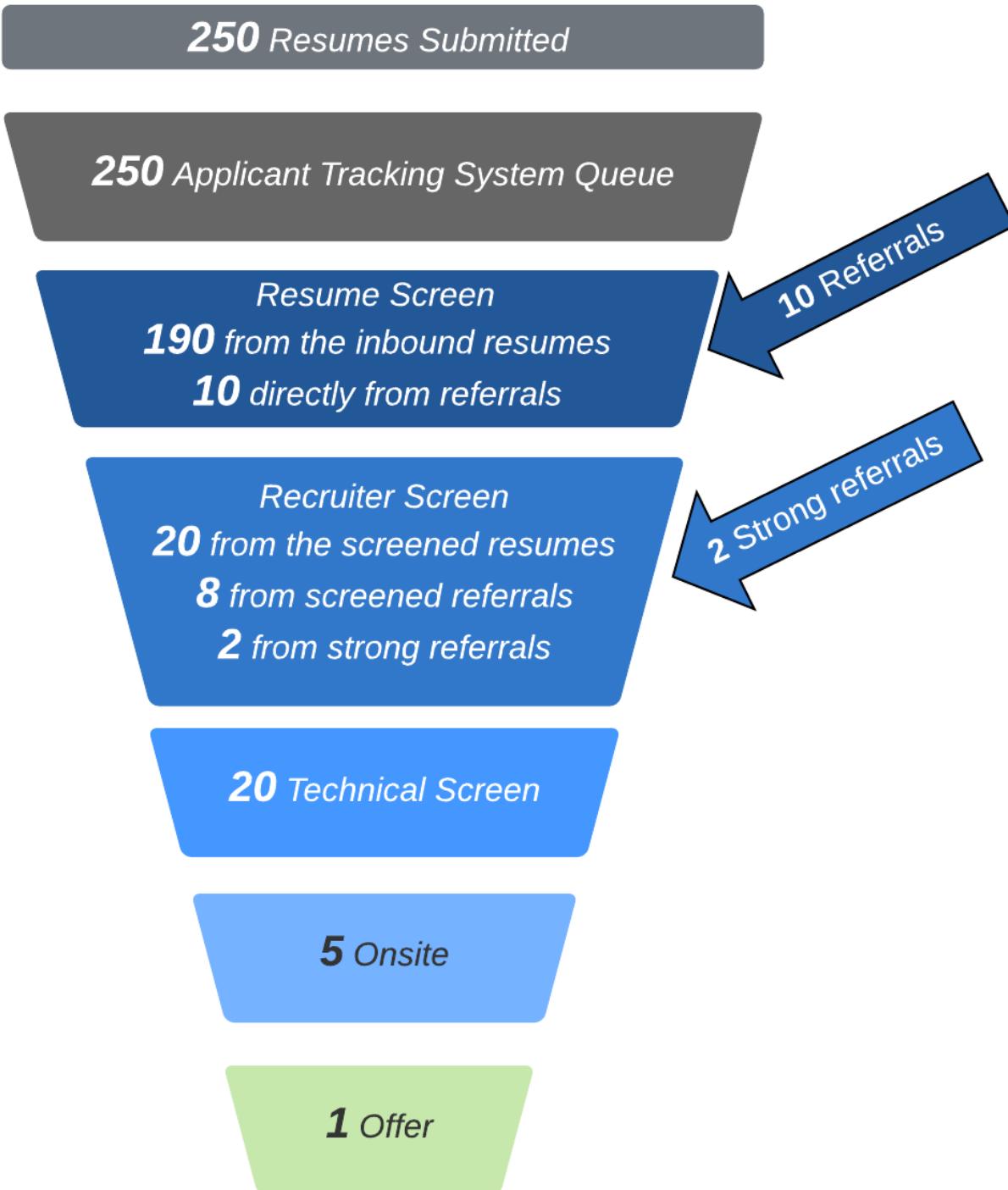
Referrals and references might sound similar, but they could not be more different. And references and background checks are sometimes confused, though they are also different.

**Referrals**—short for employee referrals—are ways for employees to refer candidates internally. Pretty much all tech companies have a referral program that incentivizes employees to make referrals and rewards them for successful referrals. When their referral is hired, employees are paid a nice bonus, often in the thousands of dollars. Referrals are treated with priority because they work remarkably well. Referrals are far more likely to go through interviews successfully, to get an offer, and to accept an offer. They are also less likely to turn out to be a bad hire who doesn't make it through their trial period. Studies like [Understanding the value of hiring through referrals](#) and [The value of hiring through employee referrals in developed countries](#) confirm this to be the case not just within tech, but for hiring in general.

**References** stand for reference checks and usually happen after the interviews but before an offer. If you have done well on the interview, many companies do reference checks to confirm that you would be a fit for the job. Companies often reach out to you, asking for you to provide these references. It's good practice to let your references know that they can expect a call. Reference checks are almost always carried out by the hiring manager.

**Background checks** are usually run after you have accepted the offer, and the offer can be withdrawn if the background check is unsatisfactory. Background checks can range in how detailed they are. They often verify your employment history for the past few years, and they could include criminal records and other official background checks. This latter one is heavily dependent on countries, states, and companies. Background checks are almost always carried out by a company specializing in these. Some industries—like finance and government sectors—are more strict on these checks than others.

Let's revisit the hiring pipeline to see how referrals change your chances of getting noticed. I'm differentiating between a very generic employee referral and a "strong" referral. A generic employee referral could be an employee working at a company you don't know personally, but after reaching out to them and asking for a referral, they referred you with your resume. A "strong" referral would be someone who works at the company who you have worked with, and they actively vouch for you.



*Illustrating how referrals can "skip the queue" in the hiring pipeline, thanks to the additional context they bring. The actual process, and numbers will vary on a role-by-role, company-by-company basis.*

Referrals almost always guarantee an in-depth resume review. In many cases, you'll also make it to the recruiter call more easily, thanks to the referral. For strong referrals, you might get a recruiter call set up straight away.

So how do you get a referral? You ask for it.

- **Check your network** to see if someone you know happens to work at the company you are applying to.
- **See if you have second-degree connections on LinkedIn**, where someone you know is connected with another person who works at the company. See if you can ask for an introduction to get a referral. When asking for an introduction, do the legwork. [The article Get Introduced To Your Sales Prospects in 4 Simple Steps](#) has solid advice not just for salespeople, but also for developers. Looking for a job is a kind of sales activity: except, you're selling yourself.
- **Cold outreaches for referrals on LinkedIn** can also work. It's less likely to work than when you have a shared connection, but it can mean a boost, compared to not having any referral. If you send a LinkedIn connection request, be sure to add a clear and concise message in the request about you looking for a referral. Consider adding an "easy way out" for the person you approach out of nowhere, noting that you understand if they can not help with your ask.
- **For well-known tech companies**, consider asking for a referral on [Blind](#). Blind is an anonymous professional network for people working at larger companies. It is common for people to ask for referrals. When you do, make sure to tag the company name, include your LinkedIn profile, describe why you are a good fit for the job, and link the job description.

If you ask for a referral, do this before you apply to the position. Most tech companies have referral systems, where employees are only eligible to refer candidates who are not yet in the hiring pipeline. This means that if you apply directly on the company's website, then ask for a referral, you often won't be able to get referred. This is because you will be considered an active candidate; your email and resume are already in the system.

It can be a delicate balance on how long you wait to apply, versus getting a referral: you'll need to make the call if you want your application to go in faster, or to wait and have it come with a referral, assuming you can get one.

### **From the inside out: what people working at tech companies say about referrals**

I reached out to hiring managers and engineers working at tech companies, asking them how important they see referrals at their company. These quotes are not representative of the company—departments might treat referrals differently within the company and policies might have changed since—but they do show that even blind referrals can help.

Facebook: "*With a referral, you are 10x as likely to get an interview. If you are a borderline hire and your referral has worked with you, and they can vouch for you technically and professionally, it can turn a 'maybe' into a 'yes'. I've seen it happen before.*"

Google: "*Almost all people I've referred, even without knowing them, have gotten an interview. Beyond that, cold referrals don't help. Referrals, where the person has worked with you and has good enough knowledge of your work can help you get an offer, though.*"

Amazon: "*A blind referral can get you a phone interview with a recruiter—this is what happened in my case, at least.*"

Microsoft: "*A referral can get you an interview faster, but it doesn't help you get an offer.*"

Uber: "*It almost certainly gets you a response from the recruiter. If the person referring you knows you well, they might also ping the recruiter directly, increasing the chance for a recruiter call.*"

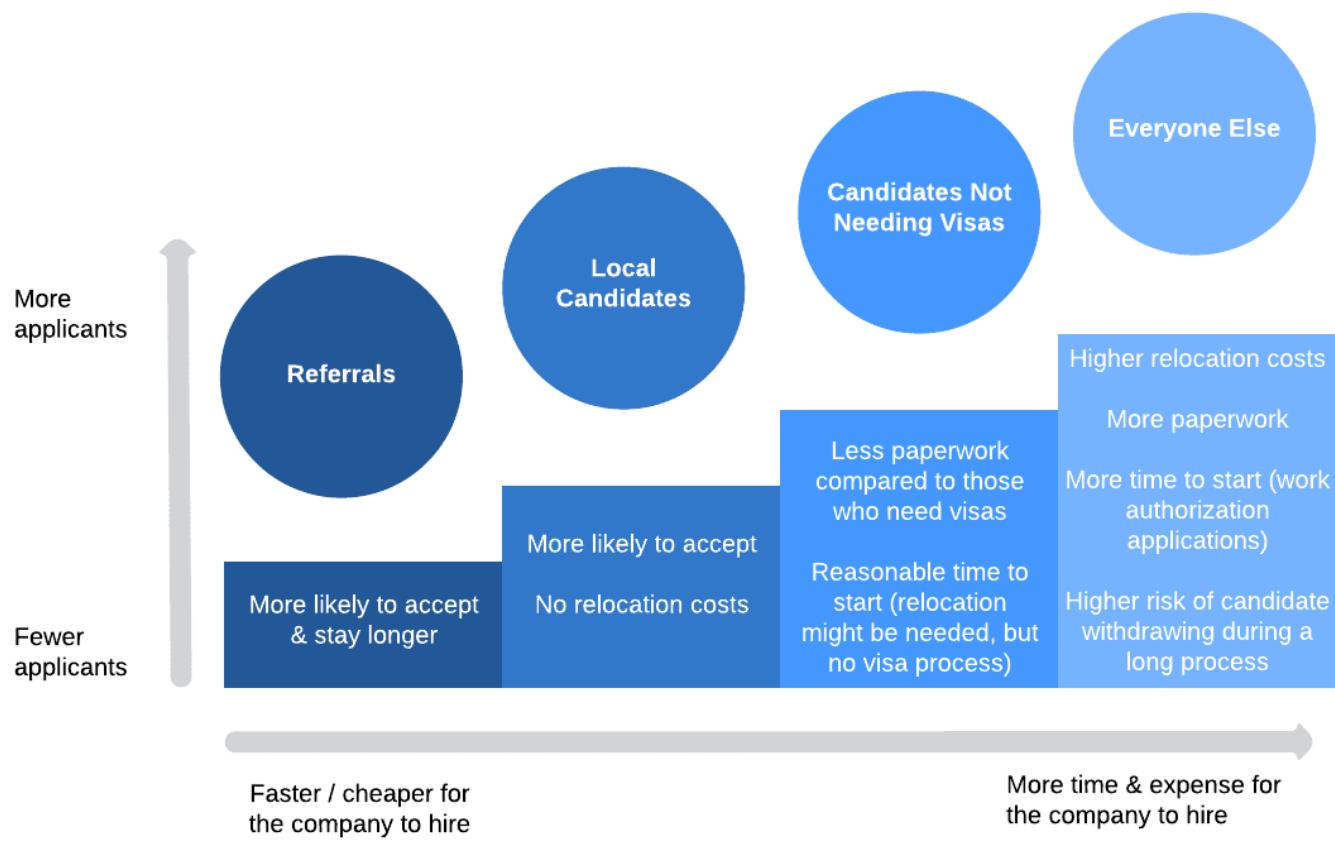
Netflix: "We usually only give referrals to people we know well—blind referrals are not a thing that I've seen."

Automattic: "A referral gets your resume and application fast-tracked to the recruiter, but that's about it."

## The Priority of Your Resume

While the hiring pipeline tells one story on how candidates go through stages, this omits an important detail: the priority of your resume, based on factors of being local and needing or not needing a visa. With a large number of inbound applications, here is how a hiring manager would often prioritize these applications.

A hiring manager's goal is to hire people who are qualified for the job as quickly and efficiently as possible. The goal of the recruiter is to help the hiring manager in this process. Given a large number of resumes with similar qualifications, how would these resumes be prioritized? Usually, they would follow this order:



*Typical priority of resumes. The more "expensive" a candidate is to hire in terms of time or money, the less priority that resume typically gets.*

1. **Referrals.** The recruiter and hiring manager almost always read through resumes that come through employee referrals in depth. While the hiring manager might have a policy of not moving forward with referrals who need a visa or relocation, this is also the most likely place where exceptions could be made, especially with a strong referral.
2. **Local candidates.** Recruiters and hiring managers perk up when seeing local candidates. Local means no paperwork to start, and no relocation is needed. There's also little risk of a candidate who accepted an offer backing out due to relocation concerns.
3. **Candidates not needing visas.** In these cases, once an offer is made, there's little paperwork needed, and there won't be visa-related delays for the person starting. They might need to relocate, but relocation would not add any uncertainties to the starting date.

- 
4. **Candidates needing visas** and relocation are usually people from abroad. This type of hire is the most time- and cost-consuming for the company. A company needs to be ready to wait for a longer visa process to go through, and pay for the additional costs, like relocation, in these cases. These kinds of hires are usually reserved either for positions that are hard to fill, or for cases where the company needs to hire a large number of people. Someone needing a visa and relocation might mean an additional 2-3 months to start the job, with an additional \$10-15K in relocation and visa costs.
  5. **Candidates from countries that are unusually difficult to get visas for**—this usually applies to US companies. In specific countries, and for certain companies, hiring developers from some countries becomes very difficult, if not impossible. One specific case is hiring for US companies—even when the office is not in the US—from countries that are on the comprehensively sanctioned countries or the targeted sanction countries list. Getting a visa can be a difficult process that few—if any—companies will want to go through.

Most companies decide on a position-by-position basis if they will go ahead with relocation and visa sponsorships. For larger companies, policies on this are usually central. Companies will typically only sponsor visas and provide relocation at senior and above software engineering levels and at roles they find hard to hire for locally.

This is even the case for the largest tech companies, like Google or Facebook. For entry-level positions such as L3, these companies also often only hire locally, filling up the headcount at these levels, making it unlikely to get visa sponsorship in the majority of cases. There are always some exceptions to this rule, based on unique circumstances. For example, at Uber, there was a point where I was able to sponsor standout L4 and L3 candidates during a hiring push - both levels below the L5, senior level.

As a rule of thumb, the more qualified local applicants there are, the less likely companies will sponsor visas and relocation. This is why it is extremely rare to get through even to a recruiter chat when applying from abroad for an entry-level position. Recruiters will follow the policy of the company and instructions from the hiring manager. They will not spend time talking to people for whom the company or the hiring manager would not sign off on making an offer.

Better companies make it clear whether they sponsor visas on the job description itself. Look out for these.

**For remote positions, the selection process is usually more straightforward.** Larger tech companies hire people as employees, and they will only hire engineers in countries where they have a legal company entity that can employ them. For example, a US-based company might have entities in the UK and Singapore, and they might sponsor hiring people to work remotely—but only in these countries. For companies like this, if you are not based in these countries or cannot move without a visa, you won't be considered for the position.

Some smaller and mid-sized companies will often hire from anywhere, also being open to hiring contractors, not just full time employees. This would mean that while you can work for these companies remotely, you will be responsible for setting up a business that the company will contract with, and you'll be responsible for all your taxes and benefits.

The downside of hiring contractors is that this setup is more fragile from the point of the company. Longer term, few companies stick with it. Things like equity become close to impossible to administer in this setup, and the working relationship needs to be a lot more transactional. Still, there is a large upside that you could be considered for the position from different companies, assuming you can work with a timezone that overlaps well enough with what the company is looking for.

---

## ***From the inside out: can I get visa sponsorship in tech when I'm a new grad or junior software engineer?***

If you are hoping to get visa sponsorship in another country from a tech company when you have little to no industry experience, as a hiring manager, I'm here to tell you the bad news: this is rarely a thing. Yes, there are a few exceptions at big companies with specialized programs that target hiring outstanding new grad talent from abroad. Still, in almost all cases, new grads won't be sponsored with visas.

Why is this? There are two reasons: the time and money cost of obtaining a visa, and the supply and demand for talent in the market.

The time and money for visa processing can be a lot of pain, especially for a small company. Money is often the smaller issue: the uncertainty of when a person can start and navigating complex visa application processes can be a real pain. I know of smaller and mid-sized companies that stopped offering to sponsor visas after they had to go through several months of paperwork and back-and-forth with the government entity for what should have been a straightforward hire.

On top of the cost of a visa, there's then supply and demand. The larger the company, the more likely it is to sponsor visas, as the more people it hires. At the same time, the more people a company hires, the more likely it has university recruitment programs and advertises its positions locally. And while large tech companies usually have a good amount of new grad and entry-level developer positions available, they can usually fill these with candidates who don't need a visa: local candidates or ones within the country.

However, with 2-3 years' experience, this changes significantly. Most companies struggle to attract developers with experience. The more senior the position, the more challenge it is to hire for these. Most medium-to-large tech companies I know of sponsor visas and relocation for senior and above positions. Some do so for levels above entry-level positions. They need to do this to hire the type of engineers they are looking for.

So my advice is to give applications abroad a shot, but don't hold your breath. Instead, focus on getting a position that is somewhat local, spend a few years working, and then you'll have many options where companies will sponsor visas.

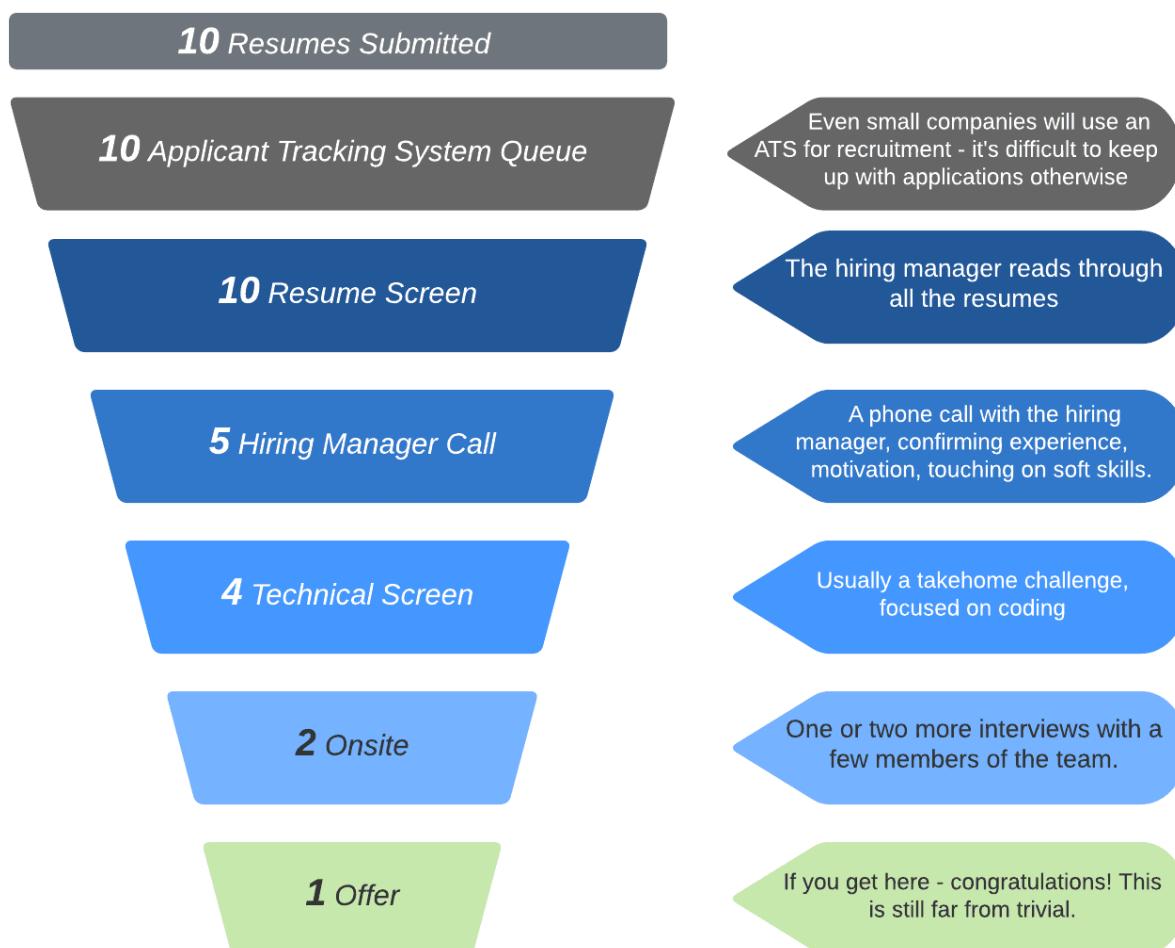
## Less Competitive Hiring Pipelines

The hiring pipeline can be very competitive for large companies. This is especially true for less senior roles. However, the good news is that not all hiring pipelines are like this. Startups, small companies and medium-sized companies differ by the number of applicants, and how involved hiring managers are with the process.

## Startup & Small Company Hiring Pipelines

Little-known startups and small companies often see few applications. This is for two reasons. First, they are often very specific in their job advert on what they are looking for, and what technology knowledge is a must. Second, they often want to hire for “hot” areas, where the large companies are also recruiting for. However, they don’t have the reach as those companies have, so they see fewer inbound applications. To offset this, they will often work with recruiting agencies to source candidates.

Startups and small company recruiting is often run by the hiring manager themselves. A typical pipeline for a non-entry level position, for a little-known company can look like this:



*A typical hiring pipeline at a startup or small company. The actual process, and numbers will vary on a role-by-role, company-by-company basis.*

---

For companies that are very small or have minimal resources to spend on hiring, the hiring manager often does all the work. This includes the resume screen, the screening call, and coordinating the onsite interviews.

**For companies with little technical expertise, you'll almost certainly have to pass algorithmic coding challenges** that are often challenging. But why would a small, barely known company have coding challenges nearly as difficult as larger tech companies have? It's because they don't have in-house expertise to interview for these skills, and they buy a screening solution instead.

If the founder is non-technical and there aren't other hands-on hiring managers, these companies will often "outsource" technical screening to an out-of-the-box solution like Hackerrank. Hackerrank and similar solutions are often built with Silicon Valley hiring in mind, meaning the bar to pass can be difficult. What's more, the non-technical hiring managers in these companies often use the score from the test as the signal on how well you did. They won't even look at your code: so a missing edge case could make you fail this screen.

Outside of major tech hubs, this kind of approach to screening software developers is more common than most people assume. It should be of little surprise that the screening and the actual work you'd do are not similar. Still, as annoying getting algorithmic challenges for a small company can be, there's an upside. Should you pass, you'll probably be one of the first hands-on technical hires. If you're passionate about hiring, you will likely be able to shape the interview process for the better, from within the company.

**For companies with technical founders, the hiring process can be one of the nicest ones.** Your resume is almost always read by a technical hiring manager, who works with developers day-to-day. The first phone call is with the hiring manager themselves—most likely, your future manager. The technical screens rarely have algorithmic challenges, and the onsite is shorter as well. There's less competition, and you are more likely to get an offer in the end. So what's the catch?

The catch is that these are small, lesser-known companies. They will almost certainly not sponsor visas. The compensation will almost certainly be below that of the large tech company. The company might seem like a really risky bet.

However, it's precisely these reasons why these companies are seeing so few applications. And some of these assumptions might not be true. The compensation might be solid—perhaps it's just not advertised. The company might still sponsor visas—especially if they are failing to attract even local talent. And while the company might be a risky bet, the risk could easily pay off.

---

### **From the inside out: how the first mobile engineer at Uber was hired**

Jordan Bonnet was the first mobile engineer hired at Uber and employee #5. He shares the surprising story of him getting this job—and an example of how applying to small companies can pay off.

*"I had been working as a software engineer for two years in France, and I wanted to get a job in the US. So I went to one of the biggest US job sites at the time, Simply Hired, and applied for more than 20 jobs. I heard back from 3 companies: Qik, Eventbrite, and UberCab. Qik told me hiring from abroad was complicated for them at the time. Eventbrite told me they don't sponsor visas.*

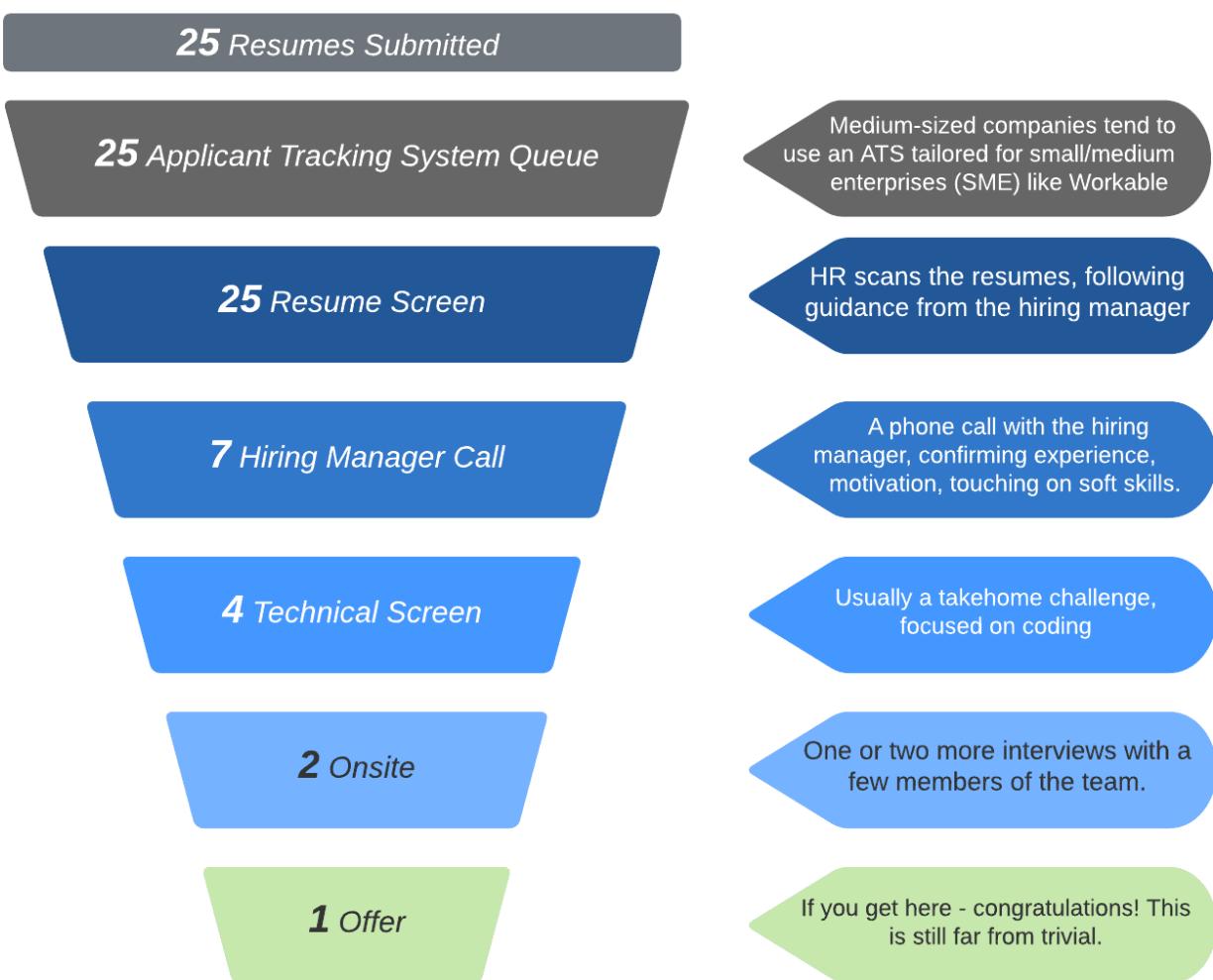
*UberCab just asked if I wanted to do the coding challenge, which I did. I then had an interview with the first engineer at the company and the CEO, and that was it. They extended an offer and sponsored my visa.*

**Applying to very small companies can be a great move** because they often have a very lightweight recruitment process and can move really fast. A couple of months into joining, I asked why they went into the trouble of hiring someone who needed a visa. They told me that they just couldn't find a mobile engineer. No one believed Uber would be big one day, and they were just too new and unproven for other developers to consider them. At the same time, I was willing to join because I had no other options."

## Mid-Sized Companies with HR Support for Recruiting

At companies that outgrow the few-person startup stage, the hiring manager realizes they can't do all the hiring work themselves. Still, the company is not large enough—or hiring fast enough—to invest in a sourcer or a dedicated recruiter. Instead, they usually hire an HR generalist with the goal of reducing the load on the hiring manager. They take on the screening the resumes and organizing of the interviews - on top of many, non-hiring related duties.

These companies are less well-known compared to large tech companies. They see more applications compared to some small companies, but nothing comparable to the large ones. The hiring pipeline will be similar to startups, but the hiring manager is now only involved later in the process:



*A typical hiring pipeline at a mid-sized company. The actual process, and numbers will vary on a role-by-role, company-by-company basis.*

The only notable difference in this process is how the hiring manager no longer personally reviews the resumes. The HR generalist is someone who is not hands-on with tech, and they will focus on instructions received. So if the hiring manager asks for 4 years or more of Java experience, they might reject resumes where this is not clear, to someone who is not technical.

## Job Aggregators

While the most well-known tech companies easily get hundreds of applications for every open position, small and relatively unknown companies struggle to get even a few applications. These companies often hire recruiter agencies to seek out candidates. And it's not just overly small companies that have this problem. Talking with recruiting agencies, it is far more common for startups who just raised a Series A to not see enough inbound job applications than most people assume.

Many of these companies spend time advertising their product—like having an article on TechCrunch—but they often don't spend nearly as much time advertising their position. They will have the position listed on the career site and perhaps pay to advertise it on a job board: but listing on a job board is already a step that many companies don't do.

**You can use the fact that many tech jobs are "invisible" to the masses to your advantage** by proactively finding these smaller, promising companies with tech positions and directly applying to them. This is a win for both you and the company. By applying early, your resume will have to stand out from fewer applications. And, assuming you are qualified and might get the job, the company will have to spend less time and money on recruitment. To do so, you need to use job aggregators to go directly to the career site postings of jobs most people are unaware of.

**Most candidates only search on job boards, which have far fewer tech listings.** Job boards are the job sites you are probably already familiar with. They have a well-known brand, and companies have to pay a hefty fee for each job advert. In return, they get a lot of eyeballs, and—hopefully—many inbound applications. The most popular job boards in tech include:

- [LinkedIn Jobs](#)
- [StackOverflow Jobs](#)
- [AngelList](#)

**Job aggregators have far more positions, as they also crawl company career pages.** If you want to find some of the “hidden gem” listings that likely get very few applications, you’ll want to use job aggregator sites. These sites will have positions that companies list directly on the site, but they also act as a “search engine for jobs”. Popular job aggregators worth browsing are:

- [Indeed](#)
- [Simply Hired](#)
- [CareerJet](#)
- [LinkUp](#)

Job aggregators are not specific to tech. You’ll have to narrow down your search criteria to the title and location, and browse through listings. When you find a position that looks interesting, try to find the company career site, and apply directly, with a tailored resume. While you could also upload your generic resume, I only suggest this if you are already getting responses. If you are struggling to get that recruiter call for jobs you seem to be qualified for, then you will see better results by sending a resume that you tweak for each job description.

Note how the first mobile engineer at Uber also found the listing via a job aggregator. Despite having raised funding from great investors, there were so few applications for the mobile engineer position, that applications got fast responses at that time. Job aggregators can help find positions with similarly small—yet potentially great—companies like UberCab in 2011.

---

## Recap

In this chapter, we've peeked behind the scenes on how the hiring pipeline works at large tech companies, startups and medium-sized companies. Every company has a different hiring process—and some can diverge from what was discussed here—but understanding what typically happens behind the scenes can help you tailor your strategy, even before submitting your resume. Here are things you can do to increase your chances.

1. **Keep in mind who the “target” for your resume is.** A recruiter, HR partner or hiring manager will be reading your CV, and you’ll want to make it clear why you are a good fit for the position—and do this in a few seconds. We will dive into tactics in Part 2 of the book.
2. **Seek out referrals**, especially for tech companies that are well-known and likely see many applicants. Referrals might be able to refer you to positions that are only internally advertised as well.
3. **Browse positions on job aggregators**, not just job boards. Job aggregators will have positions listed the earliest. Even for companies who pay to be listed on job boards, they typically list positions on their careers site first.
4. **Make a list of jobs and companies** you would consider. Add notes to how large the company is, how competitive the application process likely is, whether it is a local company, and whether you’d need a visa. Prioritize your application process based on the likelihood of you standing out. You’ll want to do some “long shot” applications, but also some “safer bets”.

## Chapter 3: Resumes and COVID-19

It's a tough time to be directly applying for software developer jobs thanks to the COVID-19 reality setting in. As of September 2020, more than 80,000 people in tech companies have been laid off, as per the [Layoffs.fyi coronavirus layoffs tracker](#),—with some of the positions eliminated being software developers. Meanwhile, there are fewer companies hiring for fewer engineering positions.

This results in what will likely be the most competitive market to land a software development job since the dotcom bust in 2001. In this market, there will be more cold applications via resumes to advertised developer positions than the decade before.

### The COVID-19 Job Market

With the pandemic, the tech job market is going through faster changes than we've seen in the decade before. More people at all levels are losing their jobs in tech and looking for new positions. Suddenly, the talent pool for local and qualified developers—from mid-level to senior and above—has increased. What used to be a developer's market is turning into an employer's market.

Until the pandemic is over, I am expecting to see companies sponsoring fewer visas, as they can now either hire qualified people locally, or advertise remote positions. For remote positions, the competition will increase multiple times, as now most developers know how—and many prefer—to work remotely.

This will make it harder for everyone to get a job. Preparation and persistence will make a difference. Even so, just having a great resume won't be enough: it is just the first step. You need to stand out with your skills and experience among other applicants. Getting into your dream companies will be tougher than it has ever been—applying to smaller and more local companies where you are the one standing out can also be a sensible strategy to pursue.

The fact that you are reading this guide shows that you are ready to put in the work for preparing. You'll also have to stay positive, focused, and be creative during this time to secure your next role.

---

## The COVID-19 Hiring Pipeline: What Hiring Managers Say

Talking with several hiring managers at small, medium and large companies, companies hiring are seeing an unusually large spike in applications. Far more people are applying to the currently open developer jobs than has been the case in previous years. Here are a few quotes from hiring managers:

*"I saw 100 applications per day for a role I posted. I work at a well-known tech company, but this is beyond anything we expected."*

*"I just looked at our ATS and we've had more than 200 applications in less than a month for the one role I'm hiring for. This is about 20-30x the normal applications rate."*

*"I've had one role open on my team for 2.5 months and I have more than 2,000 applications now. This is easily 20x what we'd normally see."*

*"We are getting 80 applications per day across 4 roles. It is getting very difficult to manage, and I've not experienced anything like this before."*

As a result of COVID-19, there are more applications for every advertised position than has been the case in more than a decade. As the number of applications for roles is at an unprecedented level, tailoring your resume for the position and catching the recruiter's attention in the first scan are all key. And referrals become the most important differentiating factor in this environment.

---

## **COVID-19 and New Grads/Interns**

If you are a new grad or someone looking for an internship, your situation in 2020 will be far more difficult than previous classes had. Companies are not only reluctant to hire, but well-known tech companies have rescinded internships and new grad offers in the first half of 2020. This is a first for most of these companies. Additionally, there's more senior talent on the market than in the decade before, due to layoffs. This means you'll be competing with more people with similar backgrounds to yourself, for fewer positions.

New grad hiring will be slow until work-from-home is the norm. Companies will be wary of onboarding people with little to no experience who they cannot coach. This means a few things. First, getting *any* job will be a great achievement and should be something to celebrate. Second, if you make it there, there will be little to no room for negotiation.

For the companies that do have junior openings, referrals become more important than before. On top of applying for the "big" companies of your dreams, apply to smaller, local companies as well—and have your resume or cover letter convey just how enthusiastic you are about *that* company.

You might need to look for alternative ways to get some experience in the meantime, like doing some open source work, building side projects or other things. I strongly recommend following the [How to be a Kickass New Software Engineer guide](#) to level yourself up while applying. The guide was written by self-taught engineer [Raymond Gan](#), who went from bootcamp graduate to senior engineer in two years. I especially recommend point 8 in the guide.

The good news is that things *will* get better. For inspiration, read how [a developer graduating into the dotcom bust turned out just fine](#). See the [Hacker News comments](#) of this article for more inspiration:

*"And even among them, the passionate people found a way. I got into tech with no education during the recession just on psychotic passion alone. I was working at <physical job> but hacking all night. Eventually I convinced someone to hire me (...) and it was game on."*

## **Experienced Engineers and The Need For Resumes During COVID-19**

Many of the developers I know have never written a resume the past decade. They almost always got their next gigs after being hunted down on LinkedIn. And if the recruiters asked for a resume, they just exported their LinkedIn profile.

With COVID-19 and companies laying off people, and far talent looking for jobs, actively applying for positions is starting to be the new reality even for relatively experienced engineers. This means having to put a resume together, and apply directly.

## PART 2: WRITING THE RESUME

Now that we understand better how your resume will go through systems and people like the Application Tracking System (ATS), recruiters or inbound sourcers, all the way to the hiring manager, it's time to get writing that CV.

How do you write a good developer resume? There are a few areas you want to get right, and the chapters cover these one by one.

[\*\*Chapter 4: Tech Resume Basics\*\*](#), we go through what recruiters and hiring managers look for in their first resume scan. We also go through several unwritten rules of resume writing.

[\*\*Chapter 5: Resume Structure\*\*](#) gives suggestions on how to structure your resume based on your work experience. Depending on whether you are out of school, have work experience, or when you are at a senior or above level, different resume structures can help you write a more relevant resume.

[\*\*Chapter 6: Standing Out\*\*](#) covers how your resume can stand out from the crowd of similar applications. We will go through examples of why being specific on your results and impact is a big differentiator, how tailoring your resume for the position is important, and we'll also touch on keyword stuffing.

[\*\*Chapter 7: Common Mistakes\*\*](#), we go through the most frequent issues with developer resumes. These improvement areas almost always result in stronger resumes—without changing the content of the resume. You'll want to steer clear of all of the mistakes we discuss in this chapter.

[\*\*Chapter 8: Different Experience Levels, Different Career Paths\*\*](#) gives tailored advice for bootcamp grads, career changers, senior engineers, tech leads and engineering managers.

[\*\*Chapter 9: Exercises to Polish Your Resume\*\*](#) is a list of activities that can help you craft an even better resume. Do these exercises to write an even more relevant one.

[\*\*Chapter 10: Beyond the Resume\*\*](#), we go through how to make your LinkedIn, GitHub and technical blogs stand out and share advice on cover letters.

Let's start off with the basics and talk about many of the unwritten rules of developer resumes.

# Chapter 4: Tech Resume Basics

There are several unwritten rules of resume writing, from not having any spelling mistakes, to listing your experience in reverse chronological order. Whenever a resume breaks one of these rules, it makes it seem less professional. And when doing the first scan on a resume, recruiters and hiring managers look for a few specific pieces of information—almost always doing this autopilot.

In this chapter, we cover what it is that recruiters and hiring managers “automatically” scan for at the first glance, and we write down the unwritten rules that resumes for tech positions should follow.

## The First Glance

Recruiters want to collect a few key pieces of information at first glance, and it is in your best interest to make this easy. If recruiters can't find this information, and there are lots of resumes to go through, they might move on to the next one. The key pieces of information are these:

1. **Years of experience.** The first thing they'll scan for is how long you have been working for. The recruiter will then mentally compare you to the internal level of the position—which is not always advertised. Say the position is for an L4 position at Facebook, Google or Uber, which is one level above the entry-level engineer—see the “Levels at different companies” table below. This is someone who is expected to have around 3-5 years’ experience, give or take. The recruiter will quickly scan to the education section to confirm your graduation date—whether this be university, bootcamp or something else—then subtract how much time has passed. If you make this information hard to find, or it's unclear, you might end up in the reject pile, the same way as if you don't have sufficient years to warrant a hire.
2. **Relevant technologies.** For the technology the position is recruiting for, how much relevant experience do you have? So e.g. if applying for a backend position for a company that mostly uses Java and Go, the recruiter would want to scan and see if they see Java or Go, and with what proficiency. This is why it's helpful to include all technologies and languages you've worked with that are *also* on the job description. If there are fewer applicants or the screener is thorough, they might go deeper and assume that you could pick these up quickly if you have several other languages: but don't count on this.
3. **Work experience.** How much relevant work experience do you have? Do you come across as someone who has *consistently* delivered impact? How was this impact measured? This is an area where quantifying things helps. If you can describe the number of daily visitors on the site you worked on, the RPS for the service you wrote, or an improvement % you've made, this is easier to translate across employers than a qualitative description of the project.
4. **Work authorization and visa status (when applying from abroad).** If your application seems like it's from abroad, do you already have work authorization? If not, what kind of visa would you need to get to be able to work for the company? Your application can seem like it's from abroad based on your contact details, the location of your last work or study experience, or even on your name. If you already have work authorization or a valid work visa, you'll want to add this clearly in any of these cases—or else the recruiter might put your resume in the “needs visa” pile, prioritizing it only after they have reviewed local candidates.
5. **Anything that clearly stands out.** Anything that pops out on the first page of your CV. For new grads, this could be your school—if it's a well known one—or an award. For more experienced people, it could be your company, a patent, a PhD, being a core contributor to a relevant open source product or something that is rare to see among the hundreds of profiles.

## Levels at different companies

When a job advert is posted, it always maps to one or more levels internally. This level information might or might not be exposed on the job listing itself. For example, a Software Engineer job listing on the Google site might have an internal mapping of levels L4 or L5 against it.

While there are no universal mappings between what each level means at each company, there are decent approximations. The Senior level at Google (L5) usually translates to L5 at Facebook, L6 at Amazon, and anywhere from 63 to 65 at Microsoft. The site [Levels.fyi](#) does a good job visualizing these levels. Keep in mind, though, that these levels are approximations, and there are no exact mappings between companies.



*Mapping of levels between several companies on Levels.fyi. The mapping is not exact, but it gives a sense of progression at each company, and how these might compare between companies.*

So how do you make the information that recruiters are looking for stand out? You make sure most information is on the first page, and you use clear formatting and good use of colors and bolding to draw attention to the relevant parts.

## Ground Rules

Your resume should be two pages or less and contain basic contact details. Use good grammar and no typos, make dates easy to read, and don't include photos or other non-required information.

There are a few things that all resumes need to follow to be considered professional-looking resumes in tech. These are the things that "go without saying"—and, because of this, they are rarely written down. Make sure your resume follows every one of these:

1. **Good grammar and NO typos.** Typos and poor grammar on a resume come across as not paying attention to detail and/or not having good command of the language. They can easily cause your resume to be ignored. Use free spell checking tools, grammar checking like [Grammarly](#), and ask someone else to re-read your CV for correctness. The same applies to punctuation: ensure this is consistent across your resume.
2. **Basic contact details.** Include your email address and relevant contact information, like phone number and the city and country where you are a resident, at the top. Keep this short and don't take up too much space with these. You don't need to add your full mailing address as contact details: no one will send you a letter in the mailbox based on your resume. Should you later get an offer, you'll be asked for all your personal details; but that's a long way ahead.
3. **Dates in reverse chronological order.** Mark your work and education experiences clearly with dates. List them with the latest one on the top, listing out earlier ones underneath.
4. **Don't include photos or non-required personal information** like your date of birth, gender, citizenship, relationship information, number of children, religion, or others. See the Avoiding Biases: Personal Details for a deep-dive on the biases this creates.
5. **Two pages or less:** this last one is not a strict rule, but a very wise one to follow. Aim to not go over this length unless you have lots—typically, beyond 8-10 years—of work experience. For new grads, fit in on one page. If you have less than a few years of experience, it's not expected you fill in the second page.

## Simplicity and Consistency

For people to read what you write, it [needs to be written well](#). This applies to resumes as well. Resumes that are simple, concise, and are easy to read will be read more. Hiring managers and recruiters will, at most, skim ones that are cluttered and overly verbose. To make your resume simple and concise, follow these principles.

1. **Clear, neat, and consistent formatting.** Use the same formatting throughout the resume. Use consistent font sizes and make the resume easy to scan through in a glance. See the templates section for pointers on good templates.
2. **Bullet points for easy readability.** Use bullet points that make the CV easier to read. Avoid paragraphs. Recruiters in tech companies are used to scanning bullet points—they are less effort to read.
  - o **Sub-bullet points: avoid.** They clutter your resume, make it more verbose, and make it harder to read. If you find yourself using these, re-edit your resume and stick with one level.
    - Using dashes for bullet points, to save space: also avoid. They look out of place and are harder to read than bullet points.
3. **Dates: use consistent and easy to read formats.** A date like “06/11–07/12” is hard to understand. The reviewer now needs to think, “is the first date June 2011 or November 2006?”. Just write “June 2011–July 2012”. Now they don’t need to think, and the year is clearly differentiated from the month. For any date span beyond a few years, you can also drop the exact month, as it becomes irrelevant, especially when it is a date that is more than four or five years ago.
4. **Use the PDF format.** Use this format and no other. Avoid formats like .doc, .rtf—they display inconsistently on machines that don’t have software like Word installed and can mess up an otherwise well-formatted resume.
5. **Be concise, and don’t spell out trivial things.** Ruthlessly edit your resume and drop sections that add little to no information. Ask yourself: “am I making a good case for why I am a good fit for this position that I am applying for?”

## Avoiding Biases: Personal Details and Photos

How would you react if a recruiter called you and told you one of the following:

- *"I'm sorry, but you're too young for this job based on your **age**."*
- *"While I'd love to proceed, we already have too many people in the office of the same **gender** and so we need to pass on you."*
- *"I have to reject you not because of your skills, but because you seem like a grumpy person based on your **photo**."*
- *"I think we should stop with the process as no one else in the office has **kids** so you wouldn't fit in."*
- *"I don't think you'd fit in with the British and Canadian people in the office, based on your **nationality**."*
- *"We like to have fun and we're all single, I'd rather not waste time with someone who is **married**."*
- *"Let's just end it here as there's no one else in the office with your **religion** and we don't want to have any arguments about this."*

Of course, you will never get a call like this: any company would find themselves in hot water if they admitted to discriminating against you on any of the above. Still, all non-essential personal information you add to your resume adds one more way that biases can kick in—either with the recruiter or the hiring manager. Adding too many personal details can result in a rejection based on bias.

Photos are never a thing for US-based positions or US-based tech companies. In tech, you don't need a photo to decide if they should move forward with you: it's about your skills, not your looks. In some countries, non-tech positions require photos, and this somehow got stuck in tech. However, all hiring managers and tech recruiters I've spoken to confirmed that photos add no value. They mentioned photos being distracting, playing to biases, doing more harm than good. If anyone really wants a photo of you, they can look at your LinkedIn profile, where you can decide whether to have one.

To visualize how biases kick in, take a look at these two resumes.

---

## Resume 1:

**GREG ROSSING**

**BACKEND DEVELOPER**

---

A result-driven backend engineer with 9 years of experience. I have hands-on experience migrating monoliths to microservices, and have a keen interest in distributed systems and modern software architecture.

## WORK HISTORY

---

**Senior backend developer**

May 2018 - present

**SkyLabs**

- Led a team of 5 engineers to migrate the booking service monolith to 7 loosely coupled microservices.

## Resume 2:

**GREG ROSSING**

**BACKEND DEVELOPER**

---

A result-driven backend engineer with 9 years of experience. I have hands-on experience migrating monoliths to microservices, and have a keen interest in distributed systems and modern software architecture.



**Nationality:** Ukraine      **Age:** 47 years  
**Marital status:** married (two young kids)

## WORK HISTORY

---

**Senior backend developer**

May 2018 - present

**SkyLabs**

- Led a team of 5 engineers to migrate the booking service monolith to 7 loosely coupled microservices.

The two resumes have exactly the same job-related content. Still, with Resume 2, many unconscious biases will kick in for any recruiter or hiring manager. Do we really want to hire someone who is this *old*? Who has two young kids and could perhaps be *distracted* from work? Who might need to relocate *with their family*? With the first resume, none of these questions even come up, and they certainly don't distract the reader. There, the recruiter and hiring manager skip straight to the skills and achievements. They decide to progress with the resume based on its contents, not the candidate's personal details.

**Do not add personal details to your resume that can lead to negative bias during the resume screen.** Biases are real, and you never know what unconscious biases you can trigger with the recruiter or hiring manager. Luckily, in tech, the criteria to get hired is based on your skills and your expertise. So do not add photos, date of birth, gender, nationality, and other details. For most resumes, you do not need more than your name and your e-mail address to apply.

### ***From the inside out: does not having a photo reduce my chances of getting hired?***

In some countries, photos are expected on CVs. Two recruiters working in these markets give their take on whether you should add photos.

Yinka Coker has previously recruited at Andela, and spent a considerable time hiring in the United Arab Emirates. Here's what he says about photos in the Middle East:

*"In the UAE and most of the Middle East, photos are expected on CVs for all roles, even tech roles. However, there is a cultural shift happening, reflecting the adoption of inclusive hiring practices. It is always better to tailor your CV to the local hiring practices in countries you apply to."*

Konstanty Sliwowski has been recruiting for nearly two decades in London, UK and Berlin, Germany. Heading up his recruitment agency, he's recruited for the likes of Delivery Hero, GetYourGuide and several high-growth startups and tech companies. Here's what he has to say about the unimportance of photos:

*"No client of ours has ever rejected a CV because the application didn't have a photo. At my agency, we do not believe a photo is necessary, despite most German companies receiving CVs with photos."*

Tech is one of the industries where not having a photo on a CV is rarely an issue. Do your research for specific countries, and reach out for local recruiters and hiring managers for specific advice.

---

## Recap: Actions to Improve Your Resume

In this chapter, we've covered why a simple and consistent resume is easier to review for recruiters and hiring managers. Spelling and following some ground rules are a must, as are not listing unnecessary personal details. To improve your resume, follow these steps:

1. **Choose a template that is simple and consistent.** Select one that is easy to scan through. Either choose this from the ones listed in the resume templates at the end of the book, or find another one online.
2. **No photo, date of birth, gender or other personal information.** Make sure you remove these pieces of information that can only create bias; that could hurt your chances.
3. **Relevant technologies and work experience on the first page.** Do you have your main languages and technologies, as well as your last few positions listed on the first page of the resume? If you don't yet have work experience, make sure your education details are on this first page.
4. **Dates of your work experience being easy to scan.** Can someone quickly tell how many years of total experience you have, looking at the resume? Make sure the dates stand out.
5. **Bullet points.** Are you using bullet points to list your experience? Make sure not to use sub-bullet points.
6. **Spell and grammar check.** Have you checked your resume not only with a spell check, but also with the free [Grammarly spell checker](#) and the also free [Hemingway Editor](#)? Make sure you fix all issues raised.
7. **PDF format and naming.** Is your final resume in a PDF format? Make sure to name it {yourname}\_resume.pdf or {yourname}\_CV.pdf, so it's easy for recruiters to identify who this one belongs to. For example, I'd call mine Gergely\_Orosz\_CV.pdf. Aim to not send resumes that showcase you having multiple versions: avoid calling it MyName\_v4.pdf, or MyName\_CompanyName.pdf.

---

## 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

When you are a student or new grad, you often feel like you have little to no experience to show. Make the most of what you have, though—and if you are truly low on experience, address this parallel to the application process. Here are the experiences that catch the eyes of people reviewing your CV the most, in priority order:

1. **Real-world work software development experience.** If you have been working part-time or full-time as a software developer, list this, together with your accomplishments. Don't be shy. Show off what you have delivered and how you have already gone above and beyond.
2. **Internships.** Be specific on results, impact, and your contribution. People with internships can stand out from the crowd or applicants. Be aware though, that for large tech companies, it is common to see applicants with multiple successful and impactful internships behind them.
3. **Sizeable contribution to real-world (open source) projects.** If you are a core or frequent contributor to a project, list this. If you have created a project that now has multiple contributors or many users, show this off. If you are not yet in these categories, perhaps you can join in with an open source project and get some experience that will help you stand out?
4. **Your school details.** Especially if the school is well-known and you have high grades, this can be impressive. If you attend a nationally well-known school, add this information to the details section. For example: "*Budapest Technology University is the #1 ranked university for Computer Science in Hungary*".
5. **Projects that stand out** due to their impact, such as their complexity, number of users or other impressive metrics.
6. **Tutoring and leadership positions in student groups.** If you have already been teaching your peers—as a teaching or lab assistant—that is a great sign of you being able to mentor others. Similarly, if you have led a student group or a project, list this. In both cases, list results, impact, and your contribution.

## A Possible New Grad Resume Structure

Here is a structure that could work well for a new grad, using the [Pragmatic Engineer's Resume template](#):

### Experience

Software Developer Intern	LogMeIn	Summer 2019
---------------------------	---------	-------------

- Internship experience summary, focusing on results, impact and your contribution

### Projects

Project Name	March 2019
--------------	------------

- Project details, specifics and a link to the project

### Education and Certifications

Degree Name, University, locations.	Expected Graduation: Dec 2020
-------------------------------------	-------------------------------

Coursework: relevant courses, like data structures & algorithms, data mining etc

Activities / Awards: ones that stand out, teaching assistant or scholarships

### Languages and Technologies

- **Languages:** List of languages you are proficient at
- **Technologies:** Same on technologies
- **Others:** Tools and concepts relevant to the job you are applying for

### Interests

- List of interests (assuming they fit on the one page)

Follow these guidelines for a resume that is easy to scan, with the most relevant parts being closer to the top:

- **Fit on one page.** As a new grad, you should fit on one page. You'll be up against hundreds of other applicants, and no one reads the second page. As you have more working experience, then a second page will start to make sense.
- **Work Experience and Internships.** If you have some work experience or internships, start with these. Having some work experience is already a great sign, and it can help you stand out.
- **Projects:** you probably don't have as much work experience, but you likely have projects to mention. Use the results, impact, and your contribution approach to describe why they are important and link to them. Where possible, link to the source code on GitHub—assuming it's nicely organized, with a good README.

- **Education.** Add details on your (expected) graduation date, details on your major, and list out any relevant or standout activities, grades, awards, and anything else that made you stand out from your peers.
- **Languages and Technologies section:** make sure to add the things you are hands-on with. This should be on the front page.
- **Interests:** it's nice to add a thing or two that you enjoy doing in your free time. These can be conversation starters on interviews later.
- **Use an order for sections that highlights your strengths.** Recruiters and hiring managers will look for the most relevant things to be towards the top of your resume—ones that indicate that you could be a good fit for the position. For most new grads, this order would often match the above order. However, you might decide to change it, when certain sections carry more weight, or are more relevant to the job you are applying for.

### ***From the inside out: how can new grads and interns grab the attention of recruiters?***

Sebastian Prieto Tovar and Claire Taylor have recruited hundreds of students and interns for Uber and other tech companies. Here's the advice they have for people starting out:

*"The resumes that stand out from the hundreds of incoming ones are the ones that are a good match for the job description and show some relevant experience. This could be internships, but it can equally be projects the person has done.*

*What we always tell students is read the job description, then amend your resume accordingly. And reach out directly to the recruiter, when you can. If you are a university student, ask for help from your career departments. They usually have lots of contacts.*

*What we look for on a resume is your studies, relevant courses, and what you're best at with your studies. Which class is a standout one that you mention? What are you most passionate about in your studies? We read through your relevant experiences in projects and internships. We also care about extracurricular activities, hackathons, working in teams outside school and apps, websites, and other cool things you created outside school.*

*For intern recruiting, we do closely look at the timing of your studies. For example, if the internship would start in June, we will only consider candidates who are ready to start at this time. Or if the job description is for people in a certain year, we look at this. Make it crystal clear on your resume that you fit the requirements: clarify when you can start, or how much time you have left on your studies.*

*Do spend plenty of time preparing for the interviews themselves. Read about the company, watch the videos, learn about the culture, read stories about employees, and understand the job description. Look up your interviewers on LinkedIn before your interview and read things that they might have published. You'll not only show that you've gone the extra mile—but you'll also learn a lot in this process."*

## 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 Farely, 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. With how the recruitment industry is going with ATS systems, one-touch-tooling, and AI, it's all about optimizing the top of the funnel to streamline the workflow. In this setup, it is key that you have a resume where a machine can identify that you have 80% of what the company is looking for."

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

Your resume should tell a story backward in time that people can glance at and understand. Take a look at this "story":

• Senior software engineer	Uber	2019–present
• Senior software engineer	Skyscanner	2018–2019
• Software engineer II	Skyscanner	2016–2018
• Software developer	Scott Logic	2014–2016
• Software developer intern	LogMeIn	2013 summer
• Software developer intern	Sensenet	2012 summer
• BSc, Mathematics	Budapest University of Technology	2010–2014

It shows progression and clarity. This is the type of clarity you ideally want to convey. Here are a few things you can do to have a clear story:

- **Make promotions clearly visible.** In the above example, it's easy to see that this person was promoted while working at Skyscanner. Recruiters and hiring managers should see it as well. Getting promoted within a company is an important signal—it shows that you can succeed in creating impact and being recognized by your company. Good hiring managers pick up on this signal, especially because, as a developer, it's often easier to job hop than get promoted within the organization.
- **Don't always stick to your "formal" titles when they describe your role poorly.** Some companies have developer titles that sound strange. Like in finance, the Associate title (for a software developer) or the Vice President title (for a senior software developer). If the position does not describe your role well, consider clarifying it and using a description that does. You can also add to it, to clarify the meaning. For example, if your title is Associate, you could write Software Developer (Associate). At a startup, I was "promoted" to take on the Community Manager role on top of being a developer. My title was officially changed to Community Manager. I still regret having my last job title on my resume saying Community Manager, instead of "Software Developer", which was my actual job—and perhaps mentioning in the description that I extended my role to include community management.
- **Dec 2013 vs 2013: drop the months** for dates that are more than a few years old. In the above example, the dates section was especially easy to scan, as there were no months added. Would have they added more information, if they were there? Hardly. When you have more than 2-3 years' experience, the people reading your resume won't care about month-level details. They are noise. So remove this.
- **Omit work experience that doesn't support your story.** I have seen people add non-technical jobs, including pizza delivery, to their profiles. I've also seen people with 15 years' experience listing all 4 internships they had well over a decade ago. Focus on your story and only leave the relevant parts. Talk about your *recent* experience much more in-depth than early ones.

---

### **From the inside out: is it ethical to change my job title on my resume?**

An experienced recruiter who has spent a decade recruiting for international startups and Fortune 100 companies shares their view on whether it is okay to change your job title:

*"The only ethical exceptions for changing your title are if your actual responsibilities in the past three or more months does not reflect your title. It is also fine to do so when your title is so company-specific that it does not transfer to the industry or other industry.*

*For example let's say you joined a startup with a software developer title. Due to the fast-paced environment you spent the last year acting as a team lead, leading a team of five. Your title has not been changed, though. Without a title change, your resume may be overlooked when applying to team lead or senior positions when recruiters skim the resumes. In these cases, I personally think that it's okay to adjust your job title to reflect your actual work done. I would not recommend it otherwise."*

As a hiring manager, my advice is similar. I know of a person who was hired as an iOS engineer to a company, and "iOS Engineer" was their title. However, three months in, they moved to backend development, and did this work for a year and a half, without a title change. For this person, I would suggest to change their title on their resume to one of "Software engineer (iOS & backend)", "Software engineer", or split out their time at the company as "iOS engineer" and "Backend engineer". Do it depending on what they'd like to highlight, and the path they intend to carry on.

## The Summary Section

Many resumes start with a section titled “Summary” or “Profile”. People often add from a sentence to a paragraph of text. Here are a few examples:

- *An experienced and positive backend developer, who enjoys working on high-performing teams and contributing to open source. Looking for a remote position as a Go developer in a finance company that values open source, hard work and gives back to the community.*
- *Solid yet versatile Software Engineer with more than 12 years experience in developing and leading projects. I hold a BSc in software engineering and am passionate about distributed systems.*
- *I am a software engineer with 5 years' experience, passionate about technology and microservices. Throughout my career I have built and managed systems from small to large in the automotive, media and restaurant industries. I have worked with product teams, implementing solutions in the cloud and working with several technologies and services.*

**Here's the thing: recruiters and hiring managers rarely read this section on the first scan,** regardless of how much time and effort you put into it. They only do so when they have decided to proceed with your resume. For less experienced candidates, with a few years' experience, I suggest to not have this section, unless you customize it for the job listing, highlighting things that showcase why you're a great fit for the position. If your resume doesn't catch the recruiter's eye, they won't read the summary section.

An overly ambitious summary section can also backfire. Say you are applying for a developer position and you say in your summary section that you are looking for opportunities to lead. If the hiring manager is not looking for someone with leadership ambitions, they might not call with you, thinking they do not intend to offer this growth opportunity. However, had you left this part out of the summary section, you might have gone through, got the position, then naturally grown into a lead on the team, over time.

**When you do have a summary section, make it short**, and add specific and practical information. For example, mention the years of experience you have, especially if this does not match up with when you graduated. Also, add highlights that showcase why you are a great fit for the position.

### Cases where the summary section can be helpful:

- **Senior/standout profiles** can greatly benefit from a summary section or a few sentences giving context on your motivation, or what you are looking for. For these profiles, hiring managers will be eager to learn more, and *will* actually read this section. For example, suppose you have four years of experience at a well-known tech company and are applying at a smaller one. In that case, the hiring manager will read the summary, where you might mention that you are looking for your next challenge within a high-growth environment, but somewhere smaller than where you currently are.
- **For fully remote positions**, a summary that also mentions that you are only looking for remote positions or that you are highly adept at remote work. This can be useful for the hiring manager.
- **When changing roles compared to your last title**, for example going back from manager to IC, use the summary section to indicate this. Again, your resume will stand out due to your last title, and the recruiter/hiring manager will be curious to know why you are applying. They will read the top-level summary—assuming you make this easy to spot.

- **With formatting that naturally lends itself to reading this section**, and when the section is clear and concise. This is usually the case with resume templates that have lots of whitespace—so the tradeoff is your resume has less information on two pages.

### **From the inside out: crafting a powerful summary section**

Randall Kanna is a senior software engineer previously at Eventbrite and Pandora. She has reviewed hundreds of developer resumes, been on hiring teams and reworked the interview process at her companies. She suggests putting in the time and effort to create a summary section tailored for the job. Here is the advice she shares in her book, [The Standout Developer](#):

*"The traditional, boring summary has become outdated: "Objective: Obtain a goal in Software engineering at a tech company."*

*Having a powerful summary section can mean you get noticed and stand out in a pile of resumes. More than that, your summary should immediately draw the recruiter in and make them want to keep reading. Show the hiring manager or recruiter who you are, and highlight an achievement.*

*The key to writing a standout summary is to do it last. Write your resume first and then, once you've got it airtight, focus on your summary section. Look for the most impressive details in your resume and articulate what you delivered. The recruiter won't really care that you are an avid gardener or reader or that you love to hike; they want to know that you'll create results for their company... period.*

**I recommend tailoring your personal summary section to the job you are applying for**, if possible. For instance, if I were applying for an iOS dev job, I would not highlight that I spent the last few years as a frontend developer. Instead, I would include that I taught myself Swift and Objective-C a few years ago. And I did it in a short period of time to step into a role my company needed.

*Finally, make sure your summary section isn't too short or too long."*

## Structure for Senior and Above People

When you have more than 10 years of experience and have worked at multiple companies, you start to stand out from the crowd of applicants. At this point, your resume will be read far more often by the hiring manager, and not just the recruiter. And hiring managers will want to understand more of your history, and what you could bring to the table.

Here are a few ways you could consider “breaking” the previously suggested principles, to play to your strengths.

**Do list your major deliverables and accomplishments—and while the two-page limit is not a hard one, try not to go over this length.** Keep your resume focused. The one-page resume is for grads, and the two-page resume for everyone else. This is the rule-of-thumb guidance for today’s tech resumes. But when you have more relevant experience to speak to, don’t necessarily be hung up on the two-page limit many guides recommend. List all your experiences that the hiring manager would find relevant—while balancing to stay concise.

**Consider having a summary section,** where you briefly describe the standout part of your experience and what the company would get with you. Tailor this one for the job. While you could consider adding details about your motivation in applying for the position, I advise against this—talking about your motivation will be part of the hiring manager interview. Note that for senior developer positions, having a summary section will rarely be a tiebreaker. The rest of your resume will be.

**Do have separate resumes for management and IC positions.** Experienced engineers can often move between tech lead, engineering manager, and senior engineering roles. And this [manager-engineer pendulum](#) is a great thing. Some of the best staff engineers I’ve worked with have been managers beforehand. However, when crafting a resume, instead of a generic engineer/manager resume, create two separate ones. One should tell the story on why you want to go back to being an engineer in your next role, the other about why you’re a great fit as a manager.

Companies rarely hire for hybrid roles. And if a hiring manager receives a resume that is a mix of a senior engineer and manager experience, with no clear story on where to go next, they will likely put it in the “unsure what to do with this one” pile. Tailor your resume and the story towards each type of position—and you’ll get callbacks far more often.

Take this example on two resumes for the same person. One is targeted at going back to an independent contributor (IC), and the other on carrying on as a manager. This person is applying for different roles and is genuinely open to both options. They do miss coding day-to-day, but they would not mind staying on the leadership track given the right opportunity.

---

**Resume 1—going back to an individual contributor role:**

# Ashley Sliden

my\_email@example.com

---

—  
Summary

I am looking to transition back to being an individual contributor senior software engineer, following several years of leading teams and heading up engineering at a startup. Though I enjoyed leading and growing teams I missed being hands-on the day-to-day. On top of my freshly sharpened coding skills, I bring strong mentorship skills, and a track record of delivering strong business results to the table.

---

—  
2018 - present

**Head of Engineering, TwoDrive**

BERLIN, GERMANY

- Started as the sole engineer, growing the team to 8 people.
  - Built and shipped the MVP of the product, and the 3 follow-up versions.
  - Setup the development environment, put best practices like CI/CD, canarying and automated unit testing in place.
  - Laid out the foundations of a loosely-coupled, event-driven architecture, built on top of AWS, Kubernetes, using Kafka and Java services.
- 

—  
2016 - 2018

**Senior Engineer, OneDrive**

AMSTERDAM, NETHERLANDS

---

## Resume 2—staying on the manager track:

# Ashley Sliden

my\_email@example.com

---

### Summary

I am an engineering manager with over 4 years experience in leading teams. I have built out the engineering organization at TwoDrive from the ground-up, building strong relationships with the C-suite. Areas I'm especially passionate about include team building, mentoring, and executing on business results.

For my next role, I am looking for opportunities where I can help scale up an engineering group.

---

### 2018 - present

#### **Head of Engineering, TwoDrive**

BERLIN, GERMANY

- Joined as the first engineering manager, hiring a multidisciplinary team of 8 engineers.
  - Put an engineering career ladder in place, promoting 30% of the engineers to the next level.
  - Collaborated with the C-suite shaping on budgeting and the framework to measure business impact of the engineering team.
  - Shipped the MVP of the product, and the 3 follow-up versions.
  - Ensured best practices like CI/CD, canarying and automated unit testing were put in place and coached engineers to take ownership of engineering quality.
- 

### 2016 - 2018

#### **Team Lead, Senior Engineer, OneDrive**

AMSTERDAM, NETHERLANDS

Note how the two resumes tell a very different story, even though all of the facts in them are 100% correct. They still belong to the same person: but this person focuses on different parts of their journey, with different end goals. A few things to highlight between the two resumes:

- **Summary tailored for the role.** In each case, the summary tells the story of why this person would be a great fit for an individual contributor or an engineering manager position.
- **Titles amended to tell the story.** The title of the person when working at OneDrive was Senior Engineer. However, in practice, they were leading a team from the first day. While the title Senior Engineer works well for the IC story, they amended this title to say "Team Lead, Senior Engineer" for the manager resume. This captures that they have not two, but more than four years of management experience.
- **Role highlights reflect on the career path.** The contents of the Head of Engineering role have little overlap between the two descriptions. This is because, on the IC resume, the person focused on their IC achievements, and for the manager resume, they highlighted the manager achievements.

**When crafting different resumes, be very-very careful not to bend facts.** Your resume will likely make its way around the company. I've heard recruiters talk about candidates who had up to 4

different versions of their resumes in the system: and the facts contradicted each other. Needless to say, this person was not invited to interviews after the team discovered that they seem to be fitting their resume to the job adverts, adding things that don't add up.

### ***From the inside out: can my resume go beyond two pages?***

Hiring managers and recruiters at different companies will have different opinions on the resume length for experienced people. They will all suggest keeping it relevant, though.

Ken Liu, who has recruited for close to 10 years at Microsoft, puts it like this:

*"Keep it concise. One to two pages is ideal. However, if you are management/director level you can go up to 3–4 pages at maximum. Though I'd add that going this long isn't desirable. Think about the fact most recruiters and hiring managers of larger companies will be reading through many CVs for this role and spend 10-15 seconds doing a first scan of the CV and making an initial judgment. This is the case even for more experienced candidates."*

Long-time CTO Steve Ball notes how listing relevant details is important, even at the expense of length:

*"I've been at the tech C-level for 20 years; I've reviewed thousands of CVs that time and hired hundreds of software engineers. It's a bit of a modern thing to have a one- or two-page CV. As a CV reviewer, I am often left without enough details to make a decision on whether to proceed. In the cases where I had a glimmer of interest, I have asked recruiters to give me a fuller fleshed-out CV from the candidate. For candidates, I recommend you mention your major deliverables and accomplishments for each role. If you've had 5–10 roles then this will easily take you over 2–3 pages. If you've had a lot of major accomplishments in a role, then list them."*

## Recap: Actions to Improve Your Resume

In this chapter, we've covered the recommended structure of your resume, based on your experience. You want to convey the most relevant details that recruiters and hiring managers care about on the first page. Telling a story, tailoring a summary section, and listing the relevant languages and technologies are all traits of well-structured resumes.

To further improve your resume, consider doing the following checks:

1. **Make sure the sections in your resume come in priority order** based on your experience level. Have the key details come first: like languages and technologies, work experience and/or education, then projects and other sections. Based on how much experience you have, you might want to switch up the order.
2. **Are you listing relevant languages and technologies?** Are you listing the technologies and languages you are comfortable with and those relevant for the role? Are you omitting ones that have no connection with it? Remember that adding an expertise level is not helpful: if anything, use the listing order to highlight ones you are more comfortable with.
3. **Are you telling a story** with your background? Do your titles show progression? Are you visualizing any promotions that you have? Have you clarified non-standard titles? Did you omit work experience that doesn't support your professional story?
4. **Is your summary section tailored to the position?** Assuming you added a summary section, is it concise enough, and does it reflect on the position? Does it talk about specifics, rather than using generic statements?
5. **Are you sharing too many or too few details on your most recent and older work experiences?** Your most recent work experience sections should have more details than those further away. Make sure that you adjust the detail to how likely it is that a hiring manager would be interested in these. For details 10-15 years ago, you can start to be concise, or even drop some.
6. **Are dates really easy to read?** Is formatting consistent across dates? Do you show the right level of detail, omitting months from years back? Can you improve the dates, making them easier to scan, without dropping key details?
7. **Is your resume length in line with your experience?** If you are starting out, can you fit everything on one page? With a few years' experience, are you fitting in two pages? For leadership positions or ones with over a decade of experience, are you happy with the length? This length should not go above 3 pages, even in cases where you have lots of relevant experience.

---

## **Chapter 6: Standing Out**

Your resume will be one of the many applications a recruiter scans through. How do you stand out from the dozens of other applications?

If you have a resume that reads very generic, you will not grab much attention. A resume where the hiring manager would have trouble recalling anything specific about your experience, even after carefully reading everything you wrote. A resume where each of your work experiences has an almost identical description of what you did: you built things with technologies and shipped projects that were supposedly important.

In this chapter, we'll cover how you can have your resume stand out by being specific about the work you did. How using numbers can make your work experience shine and grab the attention of hiring managers. We'll also go through other ways to stand out, like tailoring your resume to the job posting, and how different companies look for different things.

Let's get started by how to talk better about the impact of your work.

## Results, Impact and Your Contribution

When listing your work and project experiences, focus on what you achieved, as opposed to what you did. For the achievements, try to quantify these with the impact and (business) results. A framework you could use is “Accomplished {impact} as measured by {number} by doing {specific contribution}”. This is similar to [the structure Google encourages](#) for resumes. You don't need to use the exact same wording. However, do make the impact clear, what your contribution was, and add specifics where you can.

You want to convey that you are self-sufficient, that your work made a difference on your team, and that you are aware of your work's impact. To do so, edit your accomplishments with these points in mind:

- **Use numbers.** Quantify your impact wherever you can. Most resumes do not contain numbers: if you add these specifics, you will stand out. Instead of saying “*Built a tool widely adopted by the company*”, say “*Led a team of 3 developers to build a dependency injection framework that was adopted by 15 teams and all 50+ developers at the company*”. Numbers can be several things: number of people on the team, lines of code, code coverage % before and after, SLA changes, revenue generated by the project. They can be number of users, number of installs, number of five-star ratings, number of customer support tickets you proactively resolved, and many others.
- **Use active language** that shows what you have *done* and how you have been proactive. Use active verbs like “led”, “managed”, “drove”, “improved”, “rolled out” over passive ones like “improving” or “rolling out”.
- **Mention specific languages and technologies** that you used towards the end of your description. Impact and your contribution are more important to convey than the technologies. However, it's worth calling out what tools you've used. Mentioning technologies in this context is more powerful for hiring managers and interviewers who are reading your resume in detail. Make sure that these technologies overlap with the ones you listed in your standalone Languages & Technologies section in your resume.

**You stand out from the crowd by talking about the impact of your work and how you contributed to it, not just what you did or were responsible for.** The more senior you are, the more of an expectation this is, but doing so will make you stand out in all cases. Your resume should showcase how you have consciously and proactively added value through your actions. People who do this are the ones sought after—developers who follow directions are a dime a dozen.

## **From the inside out: how do I come up with bullet points for my resume?**

Andy Lester, is a manager, lead, software engineer and the creator of [ack](#). He wrote the book [Land the Tech Job You Love](#) after many years of and shifting through countless resumes and interviewing candidates. In this book, this is what he advises on writing bullet points that grab the attention of the hiring manager reading your resume:

*Coming up with your bullets is the toughest part of the entire résumé process. I recommend that you start it early; just scribble ideas on a sheet of paper that you carry with you. Don't worry about order or phrasing at this early phase. Write down everything you can remember doing, and worry about pruning later. It's far easier to get rid of extra information than to come up with one more bullet at the last minute. Instead, focus on the story, not specific buzzwords.*

*Résumé books often talk about the importance of using "action words" in your bullet points. They'll list pages of words you can use in your résumé to sound like you were effective: analyzed, compiled, coordinated, drafted, devised, implemented, blah blah blah. Although it's true that you want to use active verbs, as I noted earlier, **don't get hung up on which snazzy word you use. Instead, focus on the details of the work, getting as specific as possible.***

*Details include numbers. If you can quantify some value that comes from the work you did, it gives weight to the value that provided. Even if you weren't the only person working on a project, you can still discuss your involvement. Here are some examples:*

- *Increased traffic to website 50% over six months by (and list the actions you took to make it happen)*
- *Led teams of four to six programmers*
- *Created new task tracking system that reduced schedule creep by 40%*
- *Installed new routers that increased throughput 25%, virtually eliminating the 20% of help desk calls related to network responsiveness*
- *Reduced open ticket backlog from 500 to 20 in three months.*
- *Refactored codebase to take advantage of standard C++ libraries, reducing total LOC from 100,000 to 70,000*

Talking about impact and your accomplishments is one of the most underrated approaches in developer resumes. Several senior developers have come to the same conclusion:

- *"Effective resumes need to contain two things: responsibilities and accomplishments. The first tells the reader what your job was; the second, what your results were. Unfortunately, most people fail at the accomplishments part."* From the article [What accomplishments sound like on software engineering resumes](#) by developer and co-creator of Django Jacob-Kaplan Moss.
- Are you an Implementer, a Solver, or a Finder? Senior engineers are not implementers, and your resume should convey that you are a Solver or a Finder, argues software developer Itamar Turner-Trauring in the article [To get a better programming job, explain your problem-solving skills](#).
- *"Your goal should be to have at least one number in each bullet point, supporting the story that the text tells. So few resumes have any sort of numbers or statistics on them, you'll put your resume ahead of 90% of the other applicants' resumes."* in [Track your professional stats like a pro athlete to give your resume power](#), an article by manager, lead and software engineer Andy Lester, author of [Land the Tech Job You Love](#).

## Before and after: results and impact

### Before:

*Software Engineer at Gibbery*

- Worked on the Billing team, developing microservices.
- Automated error handling process of our restful API services that reduced one of our clients' support ticket rate.
- Enhanced customer experience of the MyTrinn platform through integrations of third party APIs such as zoom-rooms for remote meetings, square to handle recurring payment, etc.
- My role also comprised reverse engineering and analyzing the undocumented legacy code to understand the business logic, clean up the code to isolate different domains and then refactor it as a step towards creation of a new service that is responsible only for the billing logic

While this description is not that bad, it does not talk about specifics that someone who has not worked at the company could understand. It also had no numbers. While a colleague might know what the MyTrinn platform is, and how challenging the work could have been, the recruiter/hiring manager would have no idea. The last sentence isn't that bad, but it makes it seem like this role was assigned to the person, and the impact of this refactoring is not clear. Finally, the original description reads sloppy by closing the sentence with "etc".

### Improvement areas visualized:

*Software Engineer at Gibbery*

- Worked on the Billing team, developing microservices. (missing specifics)
- Automated error handling process of our restful API services that reduced one of our clients' support ticket rate. (missing specifics, and could describe better what you did)
- Enhanced customer experience of the MySap platform through integrations of third party APIs such as zoom-rooms for remote meetings, square to handle recurring payment, etc. (missing specifics)
- My role also comprised reverse engineering and analyzing the undocumented legacy code to understand the business logic, clean up the code to isolate different domains and then refactor it as a step towards creation of a new service that is responsible only for the billing logic (impact of work could be more clear, passive statements, description makes it seem this person was told what to do)

---

**After:**

*Software Engineer at Gibbery, Billing team*

- Developed, shipped and operated several microservices, with loads up to 200 QPS.
- Improved the availability of the receipts Go microservice from 99.8% to 99.9% by proposing, implementing and rolling out a read-through cache layer using Redis.
- Reduced support tickets by 80% for a large client by stepping up to rework error handling, ensuring 100% of error codes are mapped to appropriate HTTP codes, and non-mapped errors trigger exceptions on the Node.JS restful API.
- Improved customer conversion by 30% of the MySap co-working platform by implementing recurring payments and integrating Zoom rooms, using PHP and Go.
- Improved maintainability of the core billing engine by reverse engineering existing legacy code, isolating into different domains, and migrating to standalone services.

The “after” version adds specifics and describes what this person has done. Specifics include numbers, actions, and specific technologies. This re-edited version conveys that the person applying has moved the needle for the business, can get things done, and it mentions some specific technologies used on the project.

---

## Don't Be Humble

Your resume should sell the professional "you" for the position you are applying for. Don't claim untrue things, but do aim to paint a great picture of yourself for the audience: the recruiter and the hiring manager.

- **Talk about yourself, not your team. Avoid using "we"** and use the first person instead (in most cases, you can drop the "I"). The resume is about you, and what you have brought to the table in the past. And what the company would get by moving forward with you. Note for non-native English speakers that talking about the "we" is the norm when talking about achievements in some languages. Drop this—in an English resume you should always go with the first-person approach.
- **Be concise but not humble.** Don't hide your achievements—and when in doubt, inflating them on the borderline will hurt less than hiding them. Your CV needs to make you stand out from an already competitive crowd.
- **Make your side projects & open-source contributions shine.** If you built impressive projects or have great open source contributions, bring attention to these in those sections of your resume. Follow the results-impact-contribution model in calling this out, where you can, as opposed to just linking to your Github profile with no explanation or an app you've built. If you don't sell it, the person reading the CV might not look at it in detail.
- **Do talk about extracurricular activities adjacent to tech,** at the end of your resume. Talk about what their impact was, how they were difficult, and link to high-quality resources that you have created. For example, if you've organized a meetup with 100 participants, mention this. If you have a technical blog, link a specific, high-quality article the reader can read. Many resumes are dry, and showing off high-quality or standout activities can make a good resume even better. Be mindful of listing neutral extracurricular activities: ones that won't trigger bias. For example, when applying for a position in Manchester, UK, listing that you enjoy supporting Chelsea on their away games could trigger unconscious bias, as silly as it might sound.
- **Mention your learnings.** It is relevant to read about why and how you picked up a skill, how it impacted a current role or what you would like to achieve with this in the future. This is information that hands-on recruiters and hiring managers value hearing about, and it also makes the resume more valuable. For example, do mention if you picked up a technology to speed up a project, to stretch yourself, or other reasons. Learning new and useful skills proactively is a positive trait that hiring managers will appreciate. As a hiring manager, I wish I would see more of these learnings mentioned on resumes.
- **Don't highlight negatives** from past experiences. While you should not claim things that are not true, you don't need to list things that don't show you in a good light. Failed projects, low GPA scores and other things that come across as negative: you can safely leave these out.

## Before and after: don't be humble

### Before:

#### *Work experience*

- We developed and maintained an internal service for data serialization. The service became adopted by other teams.
- My team built a web application meant to be used as an internal tool by the DevOps team.
- Took part in Sprints plannings, helping define the technical and product strategy, and sometimes leading the meeting.
- Occasionally served as "build master" to monitor automated build processes and trouble-shoot eventual issues.

#### *Projects*

- Volunteer work with Facebook Developer Circles Cairo.
  - Managing an online user group for software developers.
  - Organizing meetups, workshops and hackathons for software developers in Ghana.
  - Building partnerships with local tech companies and communities like CodeLn, JSKongress to support the growth of the developer ecosystem.
- bareGo: This project is usually for learning about software architecture and design using the Go programming language. The concepts such as SOLID, Design patterns, etc, are applicable to other programming languages.
- RESTAPI with Node. Implementation of API design principles and concepts with Nodejs.

These are the main issues with this version:

- Work experience: much of this undersells the candidate. There are multiple references to the team over the person. It talks passively ("team member", "took part") and adds squinting modifiers that make it seem like this person had little contribution ("sometimes", "occasionally").
- Volunteer experience. In the "before" example, the candidate put quite a few details on their volunteer work: while this is not bad, it lacks data, and is verbose.
- bareGo: the description is quite casual ("this project is usually...") and it doesn't answer the question on why this project is interesting or challenging.
- RESTAPI project: this project has a very generic description. It could hide a "Hello World"-like simple project, or a complex and interesting implementation.

---

## Improvement areas visualized:

### Work experience

- We developed and maintained an internal service for data serialization. The service became adopted by other teams. (missing specifics)
- My team built a web application meant to be used as an internal tool by the DevOps team. (what was your achievement?)
- Took part in Sprints plannings, helping define the technical and product strategy, and sometimes leading the meeting. (achievements are underplayed, missing specifics)
- Occasionally served as "build master" to monitor automated build processes and trouble-shoot eventual issues. (achievements are underplayed)

### Projects

- Volunteer work with Facebook Developer Circles Cairo. (overly verbose section)
  - Managing an online user group for software developers. (passive language, missing specifics)
  - Organizing meetups, workshops and hackathons for software developers in Ghana. (passive language, missing specifics: was it one or 100 events?)
  - Building partnerships with local tech companies and communities like CodeLn, JSKongress to support the growth of the developer ecosystem. (is building partnerships strengthening the case for your application?)
- bareGo: This project is usually for learning about software architecture and design using the Go programming language. The concepts such as SOLID, Design patterns, etc, are applicable to other programming languages. (sloppy description, doesn't explain why the project is interesting or complex)
- RESTAPI with Node. Implementation of API design principles and concepts with Nodejs. (overly generic description, doesn't explain why the project is interesting or complex)

## After:

### Work experience

- Reduced duplication by stepping up to build an internal Java service for data serialization, then evangelizing it. The service was adopted by 6 other teams across multiple organizations.
- Built and rolled out an internal monitoring tool used by 20 engineers on the DevOps team. This tool alerted on 3 major outages in the first two months that would have previously gone unnoticed. Built the tool using Python, using Grafana for visualization.
- Led multiple sprint plannings on a team of 7 engineers.
- Stretched myself, teaching myself Jenkins configuration and Groovy and stepped up to be the “build master” on the team.
- Currently learning Scala on the side. After teaching myself Groovy, I became interested in learning another language built on top of the JVM. By going deep with two very different languages—a more dynamic and a more static—I expect to be able to make more pragmatic decisions when choosing languages for projects that interoperate with Java.

### Projects

- **Community Lead** at Facebook Developer Circles. I built up a local group of 500 developers, organized 30+ events and hackathons. Featured on the [Community Storytellers' Series](#).
- **bareGo**: A minimal and progressive Go framework to build efficient and scalable server-side applications. The implementation follows SOLID principles. See the [source on GitHub](#).
- **RESTAPI with Node**. Modern API design principles and concepts with Nodejs. Concepts include HATEOAS, Caching and Content Negotiation. See the [source on GitHub](#).

The “after” version addresses several issues:

- The projects mention more specifics, such as the impact, numbers, and technologies used.
- Learnings are mentioned about why this person picked up a new technology like Groovy and Scala, what they achieved and hope to achieve with it.
- The volunteer section is far more focused on what the candidate did, and what the results were. “I built up a local group of 500 developers, organized 30+ events and hackathons”. It also left out partnerships—that did not add much to the resume—and instead, pointed to press coverage that gives further credibility to this activity.
- The bareGo project now talks about the goal of the project, and the complex problems it solves, calling out the approach. As a hiring manager, I’m ready to click on it to check it out.
- RESTAPI with Node also described the project well. It adds more details on the complexity behind the project
- by listing the not-so-trivial elements like HATEOAS (Hypermedia as the Engine of Application State) and Content Negotiation. Again, this description definitely grabs any technical hiring manager’s interest.

## Write a Resume for That Job

If you have no problem getting recruiter calls with your resume, that's great news. It either means that your profile is a standout one, or that you have little competition in the roles that you apply for.

However, for most advertised positions, there are far more qualified job seekers for every tech position than there is headcount. This is great news for recruiters who are seeing lots of inbounds. However, it means you need to put in more work for your resume to grab their attention. To make your resume stand out, you need to write it for *that specific* job description.

Create a "master" version of your resume that lists out lots of details in your work experience and projects sections. Use the results, impact and contribution language. Don't worry if this version goes beyond two pages—as long as you'll be able to trim it down for each job description.

**Then, create a version of your resume for the specific job description**, re-editing it so it uses similar language to what is in the job details. Remove examples that don't help you with this position, or move them out of the way. If a job description is for an Android role with a focus on Kotlin and you have Android, Kotlin, and web experience, make sure your resume shows your Android and Kotlin contributions—and perhaps move the web experience further down.

### **From the inside out: the headhunter's advice on tailoring your resume**

Csudi Csudutov is the founder of Mimox, the biggest tech recruiting agency in Hungary. She's interviewed more than 6,000 developers in 20 years and reviewed far more resumes, developers to startups and well-known tech companies. She shares her advice on how to tailor your resume:

**"You lose very little when removing non-relevant bullet points from your 'master' resume.** Yet most are nervous to do so and opt to cram the content instead. Don't do this. Your resume is there to get you the interview. On the interview, you'll have the opportunity to talk about the various things that you did that are not on the resume. Be ruthless in removing things that don't help convey why you are a good fit for the position.

**One of the biggest 'secrets' in tailoring your resume to the job is to understand why the position was opened.** Read the job description carefully. Does it sound like they are hiring someone to build something brand new, as part of a new team, or is it more like there might be something to maintain, perhaps backfill someone? While it can be hard to tell exactly for large tech company job adverts, for small companies, here's a tip that will help.

When reading a job advert of a small company, search for the LinkedIn profile of people working there and browse through them. Do they already have someone with the technical skills that the job advertises? If yes, then you'd likely be working with this person. They will probably be on the interview loop as well. And if there's no one with the languages and technologies the job is asking for, then they are probably hiring for something new. In this case, the ability to build something from scratch might also be valued.

**Working with a headhunter or a recruiter can be beneficial to you,** as they will already have this scoop from the companies they are working for and you don't have to second guess these.

*They'll be on your side, aiming to place you to the position that is a great fit for you, and the company. So don't discount going through via agencies and headhunters for this specific reason."*

## Before and after: writing a resume for that job

Take this excerpt from a job description at Amazon. I added the highlights to point out keywords and key areas that are opportunities to mirror in your resume—assuming you do have experience in these areas. These highlighted phrases are ones that you might consider reflecting on, in your resume.

### Description

Do you want to be part of a team that designs and implements critical payment related services for Amazon with air-tight security and five-nine availability, that serve millions of requests per minute? Do you want to be part of a fun group that explores cutting edge technology, with a culture of learning from each other and developing each other? Do you want to be part of an organization that will be in the center of projects that will shape the future of the payments industry? If you answer yes to any of the questions above, this position is for you!

As a software development engineer, you will:

- Define, design, and implement multi-tier distributed services that secure and serve customers' payment data, support cool new initiatives such as mobile payment, and provide first-class customer experience on Amazon's websites and mobile devices.
- Lead the team in designing, implementing, and testing of major features in the next generation of Amazon's payments platform.
- Estimate engineering effort, plan implementation, and roll out system changes that meet requirements for functionality, performance, scalability, reliability, and adherence to development goals and principles.

### Basic Qualifications

- 4+ years' professional experience in software development.
- Computer Science fundamentals in object-oriented design
- Computer Science fundamentals in data structures, algorithm design, problem solving, and complexity analysis
- Proficiency in at least one modern programming language such as C, C++, Java, or Perl
- Effectively collaborate in a fast paced environment with multiple teams in a large organization (software development, QA, Project/Release Management, Build and Release, etc).

### Preferred Qualifications

- Knowledge of professional software engineering practices and best practices for the full software development life cycle, including coding standards, code reviews, source control management, build processes, testing, and operations
- Ability to take a project from scoping requirements through actual launch of the project
- Experience in communicating with users, other technical teams, and management to collect requirements, describe software product features, and technical designs

## Before:

### *Skills summary*

- **Languages:** Python, PHP, Java, Go
- **Databases:** Postgres, MongoDB, Redis
- **Tools:** AWS/Azure/GCP, Docker, Git, Kafka
- **Other:** data structures, algorithms, full stack software design

### *Software engineer at ThisCompany*

- Designing and developing back-end systems with different tech stack (Java, Python, Go)
- Created a generator of Grafana dashboards from microservice code
- Developed a reverse proxy for testing, caching calls to test environment
- Introduced a beta environment, where new features can be rolled out for a selected amount of customers

This is not bad—but it is clear how this description is a generic one. It does not reflect on the job description at all. Let's make it specific for the Amazon listing. Highlights mark the updated phrasing that now mirrors the job description language better. Note that the content of the resume is exactly the same. After the changes, however, it reflects the language that this specific company or job listing uses.

## After:

### *Skills summary*

- Languages: Java, Go, Python, PHP
- Databases: Postgres, MongoDB, Redis
- Tools: AWS, Azure, GCP, Docker, Kafka, Git
- Other: Data structures, algorithms, distributed systems, engineering best practices

### *Software engineer at ThisCompany*

- Defined, designed, implemented and rolled out a multi-tier customer profile service, using Java, leading a team of four engineers. This service is used by five other teams.
- Introduced best practices on reliability, monitoring and alerting. Built a Grafana generator that creates dashboards from microservice code: this tool was adopted by more than 10 teams. Improved testing practices by developing a reverse proxy for testing, caching calls to the test environment, adopted by my team.
- Improved how we do rollouts by introducing a beta environment, where new features can be rolled out for a selected amount of customers. Rolled out this environment company-wide.

The person behind the profile is still the same. However, a recruiter that reads both versions will more likely move ahead with the second, tailored version.

## Different Companies, Different Focus

Top tech companies care far less about the specific languages used, but they do care about software engineering skills. Consultancies and agencies are more interested in very specific technologies and years of experience with those technologies. Tailor your resume for each.

Depending on what type of company and what type of developer role you are applying to, recruiters and hiring managers usually pay attention to different parts in your resume. This has to do with the type of people these companies are hiring, and if the role has any specialization. The most common type of companies and roles are these.

### Tech companies hiring generalist software engineers

The “big” tech companies, and fast-growing venture-funded companies, almost always look for developers who are generalists. This is because their tech stack can be varied and change quickly. These companies look for good understanding of at least one programming language, and good knowledge of algorithms, data structures and—for senior candidates—designing systems.

A typical job description for this kind of a position could read something like this:

#### **Minimum qualifications:**

- Bachelor's degree in Computer Science, similar technical field of study or equivalent practical experience.
- Experience in coding, using either Java, C++ or Python
- Experience using HTML, JavaScript and CSS.
- Experience building up full stack features, from UI (mobile + web) to backend systems.

#### **Preferred qualifications:**

- Experience in front end development.
- Familiar with browser compatibility.
- Knowledge of JavaScript advanced features.

Software engineers working at our company develop innovative next-generation technologies. Our products need to handle information at a massive scale. We're looking for engineers who bring fresh ideas from all areas, including information retrieval, distributed computing, large-scale system design, networking and data storage, security, artificial intelligence, natural language processing, UI design and mobile; the list goes on and is growing every day.

As a software engineer, you will work on a specific project critical to our company's needs with opportunities to switch teams and projects as you and our fast-paced business grow and evolve. We need our engineers to be versatile, display leadership qualities and be enthusiastic to take on new problems across the full stack as we continue to push technology forward.

With your technical expertise you will manage project priorities, deadlines, and deliverables. You will design, develop, test, deploy, maintain, and enhance software solutions.

To grab the attention for recruiters at these companies, aim to follow these principles:

- **Do mention programming languages** you are proficient with, especially ones that the job description mentions. Knowing a few languages gives a good indication that you'll be able to pick new ones up on the job, something that is common at these places.
- **Do tailor your resume to the job description**, mentioning areas the job description asks for, assuming you are proficient with it. Data structures and algorithms, computer science fundamentals, object-oriented design, distributed systems, and anything with scale and

numbers to prove it are usually the type of experience that catch recruiters' eyes at these places.

- **Do focus on impact, and engineering metrics** of your work. Strong resumes at these places tend to mention things like RPS for systems people have built, test coverage % increases, cost savings on infrastructure, number of users, number of customer teams, latency reductions and others.
- **Do mention open source frameworks by the company** that you are familiar with, use, or contribute to. Many of these companies contribute heavily to open source, and you can stand out by being proficient users of some of these—especially when they are lesser-known frameworks.
- **Don't list too many technologies**, frameworks, tools, databases and others. At these places, hiring managers assume that you can pick up any of these quickly. Also, recruiters are far more sensitive to keyword stuffing: it reduces the value of your resume.
- **Don't list trivial tools** that require little to no engineering knowledge, or that are tied to a given technology. You'll get a frown from hiring managers when they see Trello or JIRA mentioned. You don't need to be a software engineer to know these.

### ***From the inside out: grabbing the attention of an inbound sourcer at a tech company***

Veronika Nora Nagy has recruited for Uber, as well as for startups and consultancies. She's reviewed thousands of incoming resumes at Uber when she acted as the inbound sourcer for tech positions. Here's her advice on how to grab the attention of the recruiter who first reads your resume:

*"Highlight your experience relevant for the company that you are applying for. For example, for the senior back-end roles at Uber, we were looking for engineers who have worked with large distributed systems. Engineers who have either worked with breaking down large monolithic systems into microservices, or built microservices at their current or past companies. Things around scalability and reliability and similar experiences also would grab my attention."*

***Whatever role you apply for, make sure that you clearly highlight the technologies you have worked with that are relevant for the role you are applying for.*** I also always find it a good sign when someone has worked with multiple programming languages and frameworks. It indicated that they are curious professionally, like to learn new things, and are open to new ideas.

*Keep in mind that different cultures teach CV writing in different ways—and a lot of these approaches might not work for international companies. Some cultures tend to arrange their experiences in a table format. However, recruiters at tech companies can find this hard to read. Keeping it short and concise is key.*

*As for resumes that really catch my attention: I prefer clean, easy to read resumes above anything else. Highlight your most relevant projects, the technologies you've worked with, and be ready to talk about them when you get to that recruiter call.*

*I suggest only applying to companies hiring generalist engineers if you are open to learning and using new languages, or if you are comfortable with the existing stack of the company. Take a look at the engineering blog, get a sense for the tech stack being used. It's a waste of everyone's time if you get to the recruiter call, only to tell the recruiter that you aren't interested in working with anything else than C# or Go, when the current tech stack at the company is Java or Node.js."*

## **Companies hiring for *that* specific technology**

Non-tech-first companies and smaller companies often hire for a specific technology. The tech stack at these places is already set, and very unlikely to change over the next few years. You'll be able to tell that you are looking at a *specific technologies company* from the job description that lists technologies required extensively. These technologies will be the exact stack the company works with—and the stack they are looking to hire for. Here is a typical job description for such a company:

Our company is in a phase of rapid growth, driven by increasing global demand for our services. To service this demand, we are looking to hire exceptional Java Developers to work alongside our existing team, all driving the services forward.

### **Main Duties and Responsibilities**

- Implementing the technology strategy, developing new products and services and maintaining the existing code base.
- Actively developing high quality production code for front ends and real time analytics feeds
- Delivering well-thought-out, clean code, helping Itarle scale to support business growth.

### **Required technical expertise**

- 3+ years' experience of writing high performance Java 1.8+, including concurrency and distributed systems.
- Strong knowledge and experience in unit- and integration testing, working in a CI/CD environment, and using distributed version control systems.
- Object-oriented design and reactive programming.
- Good working knowledge of Linux and Docker.
- Experience in Maven, Spring, and JMS. Gradle, Kafka is highly desirable.
- Relational databases.

### **Desirable technical skills**

- ELK stack.
- Prometheus / Grafana / OpenTracing.
- C++ / Python / Groovy / Kotlin.
- Actor frameworks (e.g. Akka).
- Javascript/HTML5.
- Atlassian workflow & productivity tools.
- Ansible.

The job descriptions for these places are also more traditional, mentioning things like "duties and responsibilities" or "technical expertise" and "delivering code". To grab the attention for recruiters for these companies, follow these principles with your resume:

- **Do mention all relevant technologies in the job description** that you are comfortable with—even if you might not be fully proficient with all of them. Many of these companies work with recruiters who are not as technical, and screen for specific technologies or phrases mentioned, as per instructions from the hiring manager.
- **Do spell out how many years' experience you have** with the main language the company is looking for. It is worth having a short summary section, listing out this information for the recruiter to see.
- **Repeat the technologies** you've used, listing them in the work experience section as well. This will confirm both to the recruiter and the hiring manager that you are hands-on with what they are looking for.
- **Do not list unrelated or trivial technologies.** While keyword stuffing is more relevant when applying for these types of companies, don't go overboard. And do remove technologies that are not relevant.

- **List relevant technology certifications** you might have. While tech companies rarely care about certifications, non-tech first companies often see these as a positive sign, assuming the person put in some effort in mastering the language or framework.
- **Do keep your resume easy to read.** Because of the many technologies you might be tempted to mention, it can be tempting to make up sentences just to be able to repeat these. Don't do this. Many applications coming into these companies are overflowing with keywords, but tell very little. Find that pragmatic line, and your resume will grab the recruiter's attention, as you have the relevant skills and a clean resume.

**Agencies** hire developers, then contract them out to client work. Unlike other companies who are hiring for a specific position, they might be more flexible when they see additional technologies that they could potentially contract out to clients. The same advice applies for applying for agencies, as it does for non-tech-first companies and smaller companies, with the exception of listing of your skills:

- **Consider mentioning all technologies** you are proficient with, not just the ones that are in the job description, when you apply for an agency.
- **List all certifications for technologies** you might have, as certifications might make you more attractive for an agency. Agencies know that they can contract people more easily if those people have certifications that prove their value.

### ***From the inside out: grabbing the attention of a recruiter for a technology-specific company***

Konstanty Sliwowski has spent close to two decades recruiting for various companies, currently specializing in Go, JavaScript, PHP and Java roles. In his guide [The Developer's Guide to Getting a New Job](#), he shares the following advice to follow when applying for roles that are geared towards a specific technology:

*"Mention your top technologies first. Another no-go is naming every single technology you've worked with. As an extreme example, there is absolutely no need to mention Microsoft Office. There are several ways you can do it. Choose to either group your skills and technologies into categories, and make an expanded list.*

*Cite examples of your skills in action. Your CV must have examples of each of your skills, as well as how, and when, and in what context you used them, and to what effect.*

*Concise job role summaries only. Three to four bullet points per role is perfect. Go for either a few bullets or a single paragraph of text covering each job summary. This normally includes your major tasks and responsibilities, and also key results. Put all the extra information on the last page. Keep sentences brief (seriously—avoid long sentences) and don't list every single tool you have worked with (or every single project you've done). Start sentences with action verbs to get points across quickly."*

## Keyword Stuffing

Keyword stuffing is a controversial but important topic. At companies where recruiters are not very technical, resume filtering is done by discarding resumes that do not have certain keywords. Recruiters and HR folks will often pattern match either based on expectations from the hiring manager or based on keywords that they have seen lead to offers in the past.

Let's take a backend job description that explicitly says that the team uses Java for development. In most cases, resumes without a mention of "Java" or "backend" would be disregarded. For a team working on distributed systems, less hands-on tech recruiters might discard resumes that don't mention anything distributed—even if they might describe working on such systems using phrases like "microservices", "messaging queues" or "globally fault-tolerant systems". The recruiter is not a software engineer: they just look for the term "distributed". Note that this practice is less common at large companies with technical recruiters—but it is a thing when screening is done by someone more junior.

A workaround is to throw all possibly relevant keywords into the resume. This is called keyword stuffing. You can see it happening with this resume, for example:

### TECHNICAL SKILLS

**Programming:** PHP, Java, JavaScript, Python (scripting experience), SQL || **Databases:** MySQL, Oracle 11g, PostgreSQL, SQLite, MongoDB. **Back-end Technologies:** Servlets, JSP, Spring MVC, Spring Boot, Laravel, Node.js, Express.js, Webservers (Apache, Tomcat, NGINX, Node) and Microservices. **Front-end Technologies:** HTML5, CSS3, jQuery, React.js, Context API, **Learning:** (React Native, Redux, Angular 7), Build Tools (Web pack, Gulp, Grunt), Babel || **Testing:** Junit 5, Jest, Mocha || **Tools/IDEs:** Visual Studio Code, Maven, Eclipse, IntelliJ, Vim, Trello, Version Control (git, GitHub, Gitlab), CI/CD (Gitlab, GitHub, Travis CI, Circle CI, Jenkins) || **API Paradigms:** REST, GraphQL || Infrastructure: AWS (EC2, S3, Amplify, Fargate), Docker, Kubernetes || **Message Queues:** Kafka, RabbitMQ || **Other Tools:** Trello, JIRA, Asana, Word, Excel

*Keyword stuffing: an example of overdoing it*

While the person doing this kind of keyword stuffing would think they've "covered" all possible technologies, it can make the resume look unprofessional. For places where recruiters are less hands-on, this approach could work, as they might "see" the keywords the job needs. For places where requirements are more clear and recruiters are technical, this strategy will work less well. Even if a recruiter would decide to proceed, as a hiring manager, I would consider putting this resume in the "maybe" pile.

So how do you include keywords relevant for the position, while also keeping your resume professional? You do this by including the most relevant keywords—technologies and frameworks—in your resume, but do this in a human-readable way.

A good way to have keywords present in your resume, while also keeping it professional is to have a short "Technologies", "Skills" or "Languages and Technologies" section where you list the technologies you are familiar with and are relevant for the job. In your Experience section, sprinkle specifics on the relevant technologies you've used in projects. It's fine to mention the same technology both in the

technologies section, as well as under the specific part of the experience. But do ensure the resume stays easy to read.

Let's go back to how to improve that part in the previous resume. We'll cut down the technologies listed to be relevant to the job description, and mention technologies that were relevant in getting certain projects done:

## **LANGUAGES AND TECHNOLOGIES**

Languages and frameworks: PHP, JavaScript, Python, Laravel, React, Angular

Technologies and tools: MySQL, MongoDB, AWS, Git

Other: Data structures and algorithms, API design, unit testing

## **WORK EXPERIENCE**

Backend Software Engineer, SmallComp

- Re-architected the restful API powering the mobile client, using PHP and Laravel.
- Improved customer conversion by 30% by building an improved checkout flow, end to end. I migrated the existing MongoDB database to MySQL, and built the frontend using React.

*Sensible keywords: cutting down to ones relevant to the position, bringing examples in the work experience section*

The result is a cleaner resume that still has key technologies listed—in fact, it reinforces the ones that the candidate has more hands-on experience with those technologies.

Recruiters will scan for different keywords for different job positions. This is another reason you'll want to create a custom resume for *that* job description. You can tailor your wording and technologies you are proficient with, so recruiters keep reading after they've seen key technologies mentioned.

Don't forget: your goal is to get through the initial resume screening where most inbound applications without referrals fail—many of whom are actually qualified for the job, but their resume does not tell the story.

## Recap: Actions to Improve Your Resume

In this chapter, we've gone through ways to have your resume stand out from the crowd. Being specific about your results and impact, and using numbers to convey the value you created is a massive differentiator.

Tailoring your resume for the specific position is even more so. Different companies do care about different areas: tech companies hiring general software engineers are more interested in seeing a breadth of skills, while companies hiring for a specific technology care more about expertise with the stack they work with. Tailoring your resume with keywords, doing sensible "keyword stuffing" is an additional step worth considering, to make sure the resume makes it through the ATS system if you don't have a referral.

To make your resume stands out, carry out the following checks.

1. **Numbers and impact in your resume.** Do you have some, or most of your results expressed with numbers or percentages? If not, aim to change this. The more specifics, and the more measurable the impact, the more you convey your results.
2. **Are you using active language?** Use active verbs that convey what *you* did, as opposed to things that happened, in a passive setting.
3. **Are you mentioning technologies in your examples?** On top of numbers, are you being specific on the tools that you used to achieve some of the results? You don't need to do this for every part, but repeating technologies that you also list on your languages and technologies section reinforces that you are hands-on with these.
4. **Have you customized your resume for that specific job, creating a specific version?** You'll want to have your resume reflect on the job you are applying for. Are you using similar language to the job description to describe your activities? Are you explicitly calling out technologies and languages in the job description that you are proficient with? Are you using similar active verbs to those the job description contains? Tailoring your resume, together with expressing the impact of your work, are the two most impactful changes you can make.
5. **Are you talking about yourself, not talking about "we" or "the team"?** Make sure your resume is about your achievements. Don't be humble—err on the side of taking credit for the work that you were involved in.
6. **Do you mention impactful, complex or interesting side projects?** Are you describing these projects in such a way that it's easy to understand why they are relevant? Don't leave off some of your additional achievements that add value to your resume.
7. **Are you applying for a generalist engineering role, or one focused on one specific technology?** Differentiate what type of position you are writing your resume for. The two types of companies should have different resumes. Tailor your resume accordingly.
8. **Have you "stuffed" keywords in a sensible way?** When comparing the job description and your resume, does your resume contain what could be the main "keywords" that recruiters will look for? Does your resume contain the key languages and technologies listed, the name of the position and the tech stack? If the position asks for a minimum number of years' experience, does your resume convey this information, directly or indirectly?

# Chapter 7: Common Mistakes

I've noticed certain repeating themes I am calling "mistakes" with resumes. These are areas of improvement, which almost always make resumes better. Let's go through these.

## Poor Format

I have seen too many resumes where developers had good and relevant experience, but this became hard to understand, due to poor choices in formatting. The most common issues are:

- **Hard-to-scan resumes.** Multi-column resume formats are far harder to scan than the one-column ones. While candidates are often proud of how much information they squeezed on one page, this format often leads to their resume being read in less detail. Simple formats work better.
- **Too much bolding.** Your resume should have little bolding, and that bolding should be consistent, limited to key parts, like dates, titles and companies. I see too many resumes bold out seemingly **random** parts in the middle of the sentence that **don't** seem to **make sense**. Focus on concise descriptions, and a clean format, and be very careful with bolding within sentences.
- **Too "flashy" resumes.** A developer resume is 95% about the content, 5% about the style. Still, some people go overboard, choosing eye-catching templates, to the point that contrasts can make things hard to read, and recruiters struggle to find relevant information. Resumes like this can work well for design- or UX-heavy positions, though.
- **Inconsistent formatting.** Different font sizes, key information being misaligned or positioning elements with spaces over tabs can all lead to resumes that show amateur formatting. While this itself will not result in a rejection for more senior candidates, it conveys that you don't have much attention to detail, and don't know how to present yourself well.
- **Sloppy phrases** scattered across the resume. For example, using "etc.", "and so on", slang or unprofessional language. Use clear and neat grammar, and full sentences.

---

## Common mistakes: bolding and inconsistent formatting

Take a look at this resume. What seems to be issues that stand out immediately?

# Rob Oldnatie

Phone: +1-123-456-7891

Software Developer

Email:[hello@example.com](mailto:hello@example.com)

LinkedIn: [in/rob432](#)

## Technical Skills

**Java, SQL, Oracle Database, PL/SQL, SQL Server, Microsoft Azure DevOps (TFS), SQL Serve, Web Technologies and Architectures (REST, JSON, XML, SOAP, HTTP), Linux, JavaScript, Python, Jira, Git, SVN, Docker.**

## Work Experience

### Software Engineer – Tax Returns

Sep 2018 – Present

Line Corp - United States

- 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%.
- Designed and developed the new **W4 interfaces** for our existing US-based clients to provide **seamless tax withholding experience** with their payroll (Workday, UltiPro etc) system.
- Redesigned the **SAP Concur interface** for our biggest client that benefitted them by **performance improvement of more than 300%**.
- Trained Developers and peer reviewed code to maintain company and industry standard.
- Technologies: Java, Oracle Database, PL/SQL, Postman, T-SQL, MS SQL Server, Azure DevOps (TFS), SQL Server Studio Management, Git.

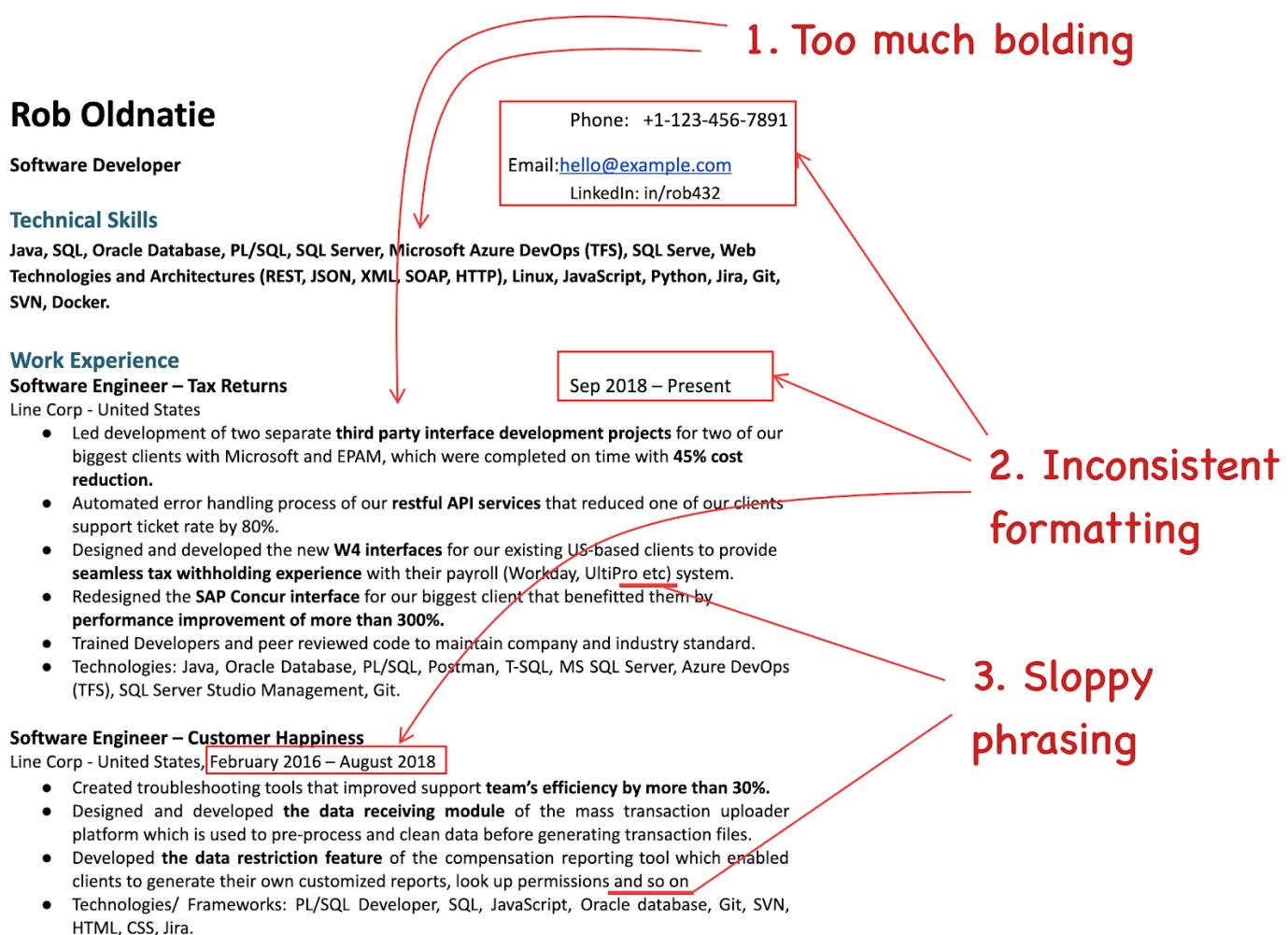
### Software Engineer – Customer Happiness

Line Corp - United States, February 2016 – August 2018

- Created troubleshooting tools that improved 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.
- Developed **the data restriction feature** of the compensation reporting tool which enabled clients to generate their own customized reports, look up permissions and so on
- Technologies/ Frameworks: PL/SQL Developer, SQL, JavaScript, Oracle database, Git, SVN, HTML, CSS, Jira.

As a hiring manager, a few things immediately pop out. First, there is far too much bolding. And the things that are bolded are inconsistent. Why is “W4 forms” bolded, when the application is for a tech company where the work has nothing to do with taxes? Second, the formatting is inconsistent. Font sizes are different, and alignment is off. Finally, the language used is sloppy in places.

Visualizing the issues:



While the content of the resume is decent, these issues convey the perception of a person who doesn't have good attention to detail. It also makes the resume harder to read and to comprehend.

## Forgetting About Your Audience

You are writing your resume for recruiters and hiring managers who you do not know. You want to convey your skills and tell a story to them, assuming they know nothing about you. Common mistakes people make include these:

- **Using internal acronyms and jargon.** Avoid using project names and acronyms that are internal and people outside the company don't understand. "*I did QA for the N94 and E70 programs.*" The recruiter reading the resume will not know what N94 was—nor would most people outside the company you worked for.
- **Not reflecting on the job, and not tailoring the resume for the position.** Make sure you keep the audience of your resume in mind. The audience is the recruiter who is recruiting for *that* specific position which you are applying for. So, say you are applying for a frontend position and have both frontend and backend experience: in your examples, always start with the frontend ones, not the other way around.
- **Using cliches over statements backed by evidence.** What does the following sentence tell about you? "*I am a team player and a fast learner who can hit the ground running.*" From the point of the recruiter, it doesn't give them any information. It doesn't contain facts, examples or specifics. It is a cliche, in its current form. Use the "results, impact and your contribution" structure in your experiences section instead to demonstrate any traits through what you have delivered. If you really think it's important to convey these, save them for a cover letter—but even there, back it up by examples. Instead of "I am a team player", let your actions speak for themselves: "I organized two team offsites and onboarded three new starters."
- **Being too verbose.** The attention of recruiters is limited: they will glance at your resume for seconds. Avoid large blocks of text or long sentences. Optimize for quick and easy reading. It's not easy to distill years of experience in a few short sentences. Still, you need to do this, and ruthlessly edit your resume.

## Unnecessary Details

Many resumes include details that are a waste of space. These are most frequently added because traditional resume templates have them, and candidates assume they should share these details. Here are the most common details that you can skip, in the vast majority of cases.

- **Photos.** Skip the photo is the strong recommendation of this book, due to [the biases it creates](#). In the rare cases in very few countries where you are *required* to add a photo, use a professional one that makes you look *good*. Smile on the photo or have a positive vibe and avoid the mugshot-like passport photos that make you look grumpy.
- **Too many contact details / social profiles.** For contact details, don't share more than four. The most common ones I see are phone, email, LinkedIn, GitHub and website. Choose no more than four of these—and list them so they take up little real estate, on the top of your resume. Anything beyond this is irrelevant. Most recruiters and hiring managers won't even click through on these ones. I've seen resumes that list Skype, Instagram, Twitter, Upwork, Medium—on top of LinkedIn, Github and a website. Stick to very few and relevant links.
- **Spoken languages.** Many resume templates come with a "Languages" section and people who are non-native English speakers keep this separate section that usually says "*Languages: Hungarian (mother tongue), English (fluent).*" This doesn't add much value—unless you're interviewing for a Hungarian company, that is. For a tech resume for an English-first company, people assume you are fluent in English and they care about your programming languages, not the different ones you speak.
- **Self-rating your skill level** on languages or technologies. Some resumes have starts, percentages, or X out of 5 scores next to languages and technologies. This self-rating can only backfire. First, if you rate yourself as 5/5 or 100%, hiring managers will be dubious. No one really knows *everything* about a programming language. However, if you rate yourself at 75% or 3 stars, other hiring managers will assume you're not that proficient. And your rating will be off, anyway. Remove all ratings, and just list things you have proficiency with. If people will want to probe this on the interview, that's fair game. But that self-rating is not necessary.
- **List of references or "references available on request".** Both are a waste of space. References will only be important after you pass the interviews. And it is expected in tech that you'll be able and willing to provide these. Large companies use background checks, and smaller ones follow up themselves, usually after asking you. But save the space from your resume for this.
- **Quotes from references praising you.** Some resumes add quotes from others, such as "*Described as 'executing at extremely high standard' on previous performance reviews*" or quotes from referrals praising you. These look out of place. Your developer resume is about you selling yourself, not adding quotes from others. It's best to save any quotes for the recommendations section on LinkedIn.

### **From the inside out: clicking through to outdated links**

Jorick Thijs Polderman, senior technical recruiter at Transferwise, mentions a common anti-pattern where candidates add links to resumes that are not up to date:

*"What I often see is that there are either no contributions or repositories on the candidate's GitHub pages, or a personal website that isn't functioning. I often see GitHub pages or personal websites*

---

*not giving away more information that could be beneficial to the candidate's application process—such as a more detailed resume or a technical blog.*

**When adding a link, make sure the links work and the information is up to date.** For example, your LinkedIn should be reflecting your resume. If you have nothing visible on GitHub, it might as well not be included. The same goes for personal websites.”

## Common mistakes: unnecessary details

Take a look at the following resume that is for an application for a position in the UK. What sections don't add much to the content?

# Pam Nitlittle

Location: Barcelona, Spain

Email: [hello@example.com](mailto:hello@example.com)

LinkedIn: [@in/pam4211](https://in/pam4211)

GitHub: [@pam4211](https://github.com/pam4211)

StackOverflow: [@pam4211](https://stackoverflow.com/users/4211/pam4211)

Medium: [@pam4211](https://medium.com/@pam4211)

Dev.to: [@pam4211](https://dev.to/pam4211)

Skype: [pam4211](https://skype.com/pam4211)



```
<!-- Usual resume content: languages & technologies, work  
experience, education -->
```

## Languages

- Turkish: mother tongue
- English: fluent
- German: intermediate
- Spanish: basic (conversational)

## References

### John McAllister - Director Engineering - HomeCorp

*"Pam is not only a very talented engineer, but also hard-working and friendly. He has collaborated very well with team mates, shipped key projects and I very strongly recommend him."*

### Sarah Smith - Senior Engineer - FlightScanner

*"Pam is the person I turned for advice and mentorship when I joined the team. He has been a great mentor, and a fantastic lead for the team. One of his biggest strengths is solving difficult problems: may these be debugging production issues, or coming up with versatile designs when building new systems."*

**Further references available upon request.**

Outside the core content, not many of the sections do add value. The photo introduces bias. There are far too many contact details and social accounts listed. There's no reason to add all of these. Spoken languages are unimportant in a market or company where English is the only language spoken. And references don't have a place on the resume, including praise. Let's remove all the unnecessary sections:

## Pam Nitlittle

Barcelona, Spain | hello@example.com | [LinkedIn](#) | [GitHub](#)

```
<!-- Usual resume content: languages & technologies, work  
experience, education -->
```

Now we only have information that is relevant for the position: the “meat” of the resume.

## Links

Links taking up too much attention (coloraturas or space), dead links, pointing to non-high-quality content.

- **Non-clickable links.** I often see links to projects, GitHub or other sites that are not clickable. The recruiter will never copy-and-paste this link, and as a hiring manager, I will also do this once in a blue moon. Make that link clickable like I did this one:  
<https://github.com/gergelyorosz>.
- **Linking to stale GitHub, LinkedIn or websites.** Many dev resumes have links to GitHub, LinkedIn or a website. While most recruiters and hiring managers won't bother clicking through, the thorough ones will. There are few more disappointing things than seeing that your GitHub is untouched for years, your pinned repos do not have READMEs or that your website was last updated 4 years ago. Link only if they are up to date. Otherwise, save the space and remove it. Contrary to popular belief, you don't need to have these links on your resume—especially when their contents don't add value to your current application.
- **Leaving in the full URL of the link.** In some resumes, links to portfolios or projects are pasted in their full length, such as  
<https://github.com/gergelyorosz/PythonGettingStarted/blob/master/README.md>  
Don't do this. Make links clickable, and hide the full URL behind a name that describes what the link is for. No one will print out your resume, then type in the URL into a browser.
- **Links standing out too much in color and style.** Many resumes leave links as they are, with the [default blue color](#). While this would not be a problem, this is often the only color in the resume, pulling eyes to links that are meant to be details. Consider making links blend in, rather than standing out, by keeping them the same color as text, and [underlining them](#).

## Common mistakes: links

Take a look at a section of this resume, which contains several links.

### Sam Notohere

E-Mail: [hello@example.com](mailto:hello@example.com)  
Website: <https://SamDev.github.io/>

## WORK EXPERIENCE

<b>Cloudless</b> , Backend Software Engineer(Remote)	Jan 2019 – present
• Re-architected and maintaining a RESTful API using PHP/Laravel with third-party API integrations such as zoom-rooms to enrich customer experience.	
<b>MennoMark</b> , Team Lead	Mar 2018 – Jan 2019
• Managed a team of three developers on the development team. • Trained and on-board two new software developers on the software products( <a href="#">MennoMarker</a> , <a href="#">MennoMarket</a> ). • Acted in leadership capacity and worked together with the rest of the development team to deliver quality software. • Built a mobile app <a href="http://www.mennoInova.com/apps/farming/download">http://www.mennoInova.com/apps/farming/download</a> for educating farmers enrolled in the Farmer Business Management programme. This was built with Kotlin, and SQLite Database. • Designed and implemented an API for interfacing the MennoInnova mobile application using PHP/Lumen, AWS(RDS, Elasticbeanstalk), PostMan.	

## PERSONAL PROJECTS

### pythonGettingStarted

This project is usually for learning about Python concepts, with example code.

<https://github.com/gergelyorosz/PythonGettingStarted/blob/master/README.md>

### restAPI with Node

Implementation of modern API design principles and concepts with Nodejs. Concepts include HATEOAS, Caching, Content Negotiation, etc.

<https://github.com/gergelyorosz/restApiWithNode>

This resume section contains most of the common mistakes with links. The color of them stands out and draws attention to the links instead of the relevant content. The links take up too much space by typing out the full URL. The font for the links stands out—there is little reason to use a distinct font. Let's address these issues. Note that there's a lot more to improve on in the resume content. In this example, we'll only improve links and highlighting.

# Sam Notohere

hello@example.com, [SamDev.github.io](#)

## WORK EXPERIENCE

---

**Backend Software Engineer** Jan 2019 – present

Cloudless

- Re-architected and maintaining a RESTful API using PHP/Laravel with third-party API integrations such as zoom-rooms to enrich customer experience.

**Team Lead** Mar 2018 – Jan 2019

MennoMark

- Managed a team of three developers on the development team.
- Trained and on-board two new software developers on the software products ([MennoMarker](#), [MennoMarket](#)).
- Acted in leadership capacity and worked together with the rest of the development team to deliver quality software.
- Built a [mobile app](#) for educating farmers enrolled in the Farmer Business Management programme. This was built with Kotlin, and SQLite Database.
- Designed and implemented an API for interfacing the MennoInnova mobile application using PHP/Lumen, AWS(RDS, Elasticbeanstalk), PostMan.

## PERSONAL PROJECTS

---

### [pythonGettingStarted](#)

This project is usually for learning about Python concepts, with example code. See the code [on Github](#).

### [restAPI with Node](#)

Implementation of modern API design principles and concepts with Nodejs. Concepts include HATEOAS, Caching, Content Negotiation, etc. See the code [on GitHub](#).

With the changes, we've removed the highlight colors for the links. Now, these don't draw any attention. Instead, we can focus on drawing attention to where we'd like the recruiter and hiring manager to look first. In this case, this person chose to focus on the titles, the dates of employment, and the names of the personal projects. There's also an option to use a highlight color to make the resume even more readable, for example, by coloring the parts that are just section names with a highlight color:

# Sam Notohere

hello@example.com, [SamDev.github.io](https://github.com/SamDev)

## WORK EXPERIENCE

**Backend Software Engineer** Jan 2019 - present

Cloudless

- Re-architected and maintaining a RESTful API using PHP/Laravel with third-party API integrations such as zoom-rooms to enrich customer experience.

**Team Lead** Mar 2018 - Jan 2019

MennoMark

- Managed a team of three developers on the development team.
- Trained and on-board two new software developers on the software products ([MennoMarker](#), [MennoMarket](#)).
- Acted in leadership capacity and worked together with the rest of the development team to deliver quality software.
- Built a [mobile app](#) for educating farmers enrolled in the Farmer Business Management programme. This was built with Kotlin, and SQLite Database.
- Designed and implemented an API for interfacing the MennoInnova mobile application using PHP/Lumen, AWS(RDS, Elasticbeanstalk), PostMan.

## PERSONAL PROJECTS

### **pythonGettingStarted**

This project is usually for learning about Python concepts, with example code. See the code [on Github](#).

### **restAPI with Node**

Implementation of modern API design principles and concepts with Nodejs. Concepts include HATEOAS, Caching, Content Negotiation, etc. See the code [on GitHub](#).

We've gone from having the default link color highlighting parts of the resume that were unimportant, to the important parts of the resume standing out. Now the "placeholder" information stands out with a color of choice and makes the resume easier to scan. Links are still clickable, and an underline invites people to click them. In practice, recruiters and hiring managers will only click these links after finding the key information they are looking for in the first scan.

---

## Recap: Actions to Improve Your Resume

In this chapter, we've looked at some of the most common mistakes with developer resumes. Poor format with sloppy phrasing and forgetting about who you are writing for—the recruiter and the hiring manager for *that* specific position—are common ones. Adding unnecessary details, from personal details to spoken languages or a list of references is another one. And drawing the reader's attention to things that are not really important—for example, by using links that are long and have a standout color—is also something that you'll want to avoid.

Do the following checks to ensure your resume does not have a "classic" mistake that will make a recruiter roll their eyes:

1. **Do you have a photo on your resume?** Remove it.
2. **How many contact details and social profile links do you have on your resume?** You should have no more than four. Do you need all of these?
3. **Do all the links work on your resume?** Click through each of them.
4. **Do all the links point to professional pages on your resume?** When you click through, will people see well-formatted, well-written content that conveys the type of work they can expect from you? Ensure this is the case also for your blog, GitHub, LinkedIn and any other links you provide.
5. **Is your resume easy to scan?** Give your resume to a friend and ask them to take a look at it for 5 seconds, then tell you their immediate impression. Did they see key details? Did they think it's too crowded? Or that it has too little information?
6. **Is your resume simple enough?** Are you using any "fancy" designs? If so, is this kind of "fanciness" getting in the way of readability? If it's not in the way of readability, that's great. However, if it's making things harder to scan, consider going with something more simple. Note that for frontend, UX or design-heavy positions, some nice flair can be a great touch.
7. **Is the resume concise and to the point?** Are you using short sentences? Do you have bullet points with rarely more than two lines?
8. **Are you bolding too much?** Aim to not bold anything **inside** a sentence, like in this one. Only bold important things like job title, company name, date, or headers.
9. **Is formatting consistent?** Are you using standard font sizes, and aligning with tabs, over spacing things to be approximately close?
10. **Are you using internal company jargon?** Do you have project names that only people inside the company would understand? If so, consider changing them to describing the project.
11. **Do you have a (spoken) languages or a references section?** You can probably remove both of these.
12. **Do links stand out with a different color or formatting?** They shouldn't. They take away space and attention.

# Chapter 8: Different Experience Levels, Different Career Paths

Depending on how much experience you have had, you'll want to focus on different areas of your resume. In this chapter, we'll share specific advice for students, bootcamp grads, career changers, senior and above engineers, tech leads, and engineering managers.

## Current college and university students

If you are still in school, you are already contributing to your future resume. On top of mastering software development, software engineering and computer science fundamentals, here are a few things you can consider that should help with your future job search.

- **Apply for internships.** One of the best experiences you can have as a student is working as an intern, on real-world problems, in an industry setting. To do so, seek out internships. You usually have to apply well ahead of time, and do your preparation for the interviews. Internships are something I did not do while at college—and I wish I had.
- **Volunteer for extracurricular campus activities.** You will likely have opportunities to help out professors, lead laboratory practices, organize events, and others. Consider stepping up for these. You'll gain hands-on experience in mentoring others, leading groups, and taking initiative.
- **Publish your university projects.** You'll undoubtedly have class projects to complete, some involving building working software. Consider putting in some extra effort and publishing these online, for others to be able to download or try. This is especially true for your final projects, which will be more complex than usual.

All of these activities will help you grow faster—and these additional experiences can help you stand out when looking for jobs as you graduate. Internships can be a lot more than this, as you might get return offers from when you graduate.

## Bootcamp grads

As a bootcamp grad, your experience will hardly stand out. While you likely have learned basic software development skills, your time spent studying these concepts will be less than college or university graduates studying software engineering or computer science. You will, most likely, also be competing with several other bootcamp grads for the same position.

As a bootcamp grad with no professional software development experience, you will be able to apply to positions where hiring managers are okay with someone with no software development experience. Unfortunately, not all jobs will not be in this category.

## Projects

Projects are one of the key areas where you can stand out and differentiate yourself. Most bootcamp grads will have projects listed that they did as part of their bootcamp course. To stand out, you'll want to showcase better examples. Consider the following to stand out.

**1. Publish projects with good READMEs** on your Github page. Aim for the README to have a good summary, screenshots, details on how to run, and how to test. Take inspiration from [projects with awesome READMEs](#).

**2. Have tests for your project that can be run.** Most bootcamp grad projects don't have automated tests, such as unit tests or end-to-end tests. Many of them have not heard of TDD (Test Driven Development). To learn more about this, see the [Learn Test Driven Development](#) resource.

If you have tests in your project, add a "Running tests" section on your README, to make it clear how anyone can run these. For an example on how to document running tests, see [how this is done in Leaflet](#).

Once you have tests in your code, call this out on your projects. In the project description, mention the code coverage percentage—and see if you can get this higher. Having built a project that is tested will already help you stand out from the crowd.

**3. Contribute to other open source projects on GitHub.** If you get to merge your pull request, you can claim that you've contributed to a project used by more than a certain number of developers. Stars on the project is a metric you can use here. For example, for a project with 28,000 stars, you can state: *Contributed to the Leaflet, the leading open-source Javascript mobile friendly-interactive library with more than 28,000 developers using it.*

**4. Turn takehome projects and coding challenges into fully-fledged projects.** Assuming you make it through the resume screen, a next step will be a coding challenge you need to submit. After submitting this challenge, don't stop there. Continue the work and publish an even more polished solution online, with code on Github, tests in place, and a nice user experience. Consider spending a little money for a domain and hosting, and have this project be accessible to anyone—and add it to your resume.

If your takehome project doesn't take you to the next round, ask for feedback, and build that feedback into this project. Once you publish a project you're proud of, you can always send a follow-up to the company, showing them the improved version you've built. Don't hold your breath—but who knows?

## **From the inside out: how can I stand out as a bootcamp grad?**

Raymond Gan graduated from Flatiron School in 2013, landed a job as a software engineer in New York after graduation, and has worked at several startups and tech companies as a software engineer since. He runs a study group for bootcamp grads, and here is his advice on standing out from other applicants:

*"The most valuable thing you can do to get jobs, besides paid software work, is to fix bugs on/add features to popular open source projects on GitHub! It's more valuable than LeetCode problems or student projects (which have NO customers).*

*Almost ZERO junior people I've known in 6 years have made pull requests to popular open source projects and gotten them merged into a master branch! ZERO. Open source work is more practical, impressive, and directly related to our daily jobs. It has REAL CUSTOMERS! That's the work companies most want!*

Here is a list of [beginner-friendly projects to get started on](#). Or you could look at popular projects with hundreds or thousands of stars, and their open issues.

Take the [Yarn package manager](#). Based on the number of stars, at least 35,000 developers use this tool. So if you get your pull request merged, it means senior JavaScript engineers think your code is good enough for 41,000 people. What impresses a company more? Open source work, LeetCode problems, or yet another student project?

For inspiration on pull requests to start with, see [a collection of good first issues merged](#) for the [Leaflet](#) project. Leaflet is a library for mobile-friendly interactive maps. All of these are small bug fixes—read through them to see how small a fix can be.

Continue your education through making pull requests for active projects. When you make a pull request to master, it will probably get rejected. So what? Those who reject your code will be senior engineers who are owners of these projects, and they'll often leave you comments to improve your code before they will merge it to master: "Add tests. Fix this bug here. What if this bad thing happens? What about using this algorithm?"

## Before and after: projects

Most resume advice in this book shows how to change the phrasing of your experience. This example is different. It shows what happens when you grow your experience and polish your projects while applying for jobs.

As a bootcamp grad, it's hard to stand out with the basic bootcamp projects. However, if you invest time during your job search to improve your projects—for example, by adding tests—, contribute to open source projects, or to do a paid project on a freelancing site, this can make a large difference.

### Before:

#### Projects

- Reversible (react / redux / express)  
A full-stack app to fetch events from an API and allow users to add them to their personal goals.
- Todoist (react / redux)  
A todo app for people to record and keep track of their goals

### After:

#### Projects

- **Leaflet** (JavaScript). Contributor to the leading open-source JavaScript library for mobile-friendly interactive maps, used by more than 28,000 developers. Contributions include adding tests and GridLayer async mode.
- **LightningFastAndCuteImageSearch.com** (JavaScript, React, Jest, Mocha, AWS). A unique take on image search, combining multiple image sources in performant experience. The site is open-source, fully tested and has received praise on Reddit. See the code on GitHub.
- **Insurance comparison tool** (JavaScript, React, Redux, TailwindCSS). Built a tool used by 500 businesses. Contracted with a niche insurance provider to compare small-business insurances. See more details here.
- **Data structures in JavaScript**. My robust and tested implementation of the most popular data structures like hashtable, heap, queue, stack, B-tree and others. 100% code coverage. See the code on Github.
- **Reversible** (JavaScript, React, Redux, Express, Jest, Mocha). A fully-tested full-stack app with 100% code coverage. See the code on Github.

What happened between these two versions? Five months, actually, while this person was job hunting and growing their skills. Here's how the additional projects came about.

1. **Contributed to popular open source projects.** This person looked through the list of beginner-friendly open source projects, filtering to their strongest language, JavaScript. After finding ones that had the most beginner-friendly issues, they spent time playing with the project, reading past pull requests, and finally, making some first changes. This was a lot of back-and-forths, but lots of learnings. This open source contribution now stands out and the person is more confident in their skills.

---

**2. Published one of their takehome challenges as a fully-fledged project.** One of the takehome challenges was building an image search app that uses the Flickr API. They completed the challenge—though they did not get to the next interview stage, unfortunately. Still, this person took this project as an opportunity to turn it into a production-ready product, with more features, unit and UI tests, and integrated Flickr, Giphy, and Unsplash image sources.

They also bought a domain to run the service on to make it stand out. The domain cost \$15—a good investment for the job search. They made it look and feel professional, tested it with their friends, and submitted the finished work and code to a Reddit forum for feedback.

**3. Landed a paid project on a freelancing site.** While job searching, this person responded to multiple projects on freelancing sites. They ended up getting a project with an insurance comparison small business. The project was difficult, and the client argued about the price. As this person's main goal was to get experience, they decided to not care about making a profit. Instead, they delivered high-quality work.

They published screenshots and code snippets of the work on their portfolio page afterward. The nice thing about paid projects is how real customers use them. Also, you don't have to disclose how much—or how little—you got paid.

**4. Turned one of their coding challenge interview learnings into a public project.** This person struggled with a few of the coding challenges, especially when it came to using data structures. So they decided to practice them in public and implemented several data types, with tests, and published them with a tidy README. Not only did they learn these structures better, they now have a project to show for it!

I did exactly this on an earlier job search with [SwiftAlgorithmsAndDataStructures](#). You can also find several other examples on implementing the most popular data structures and algorithms in a given language, for example [using JavaScript](#) and hundreds of others [on GitHub](#).

---

## Career Changers

When you've changed careers from another profession to software development, most of the advice for bootcamp resumes apply—assuming you have some kind of bootcamp training behind you. Regardless if you have it or not, you'll have lots of past work experience—that will likely have little to do with software development.

Here's advice to put a good resume together.

- **Have a summary that explains that you've changed careers** and your motivation. It's helpful for recruiters and hiring managers to have the context that they're reading a resume of someone who has recently changed into software development.
- **List languages, projects, or portfolio** towards the top of your resume that show that you're a hands-on software developer. Aim to link working projects and source code, where you can.
- **Link to working projects and source code** where you can. As a career changer with no years-long university education, you'll need to have projects to prove to the hiring manager that you know how to build software, and an interview would not be a waste of time for them. Follow the advice listed for bootcamp grads to beef up the projects you can present.
- **For your past work experience**, be concise in summarizing it. Have a one-sentence summary on the job itself, and focus any additional bullet points on skills that could be transferable to software development. Proactively learning about new things, teamwork, mentoring, or public speaking could be good examples.
- **Have a short resume.** Try to fit on one page—and don't go beyond one and a half pages. You might be itching to share more about your past positions, but the recruiters and hiring managers won't care much. They'll be far more interested in your standout projects and any relevant experience you might have. Similar to how new grads rarely go over one page, aim to do the same.

An example career changer resume is this one:

# Tom Clintts

resume@pragmaticengineer.com • New York • [LinkedIn](#) • [Github](#)

I am a motivated software developer, changing careers after many years in the finance and management fields. I'm proficient with the MERN stack, am a fast learner, and I'm enjoying every bit of the software development.

## Languages and Technologies

- Languages and Frameworks: JavaScript, React, Redux, Node.js, Express.js, Jeft, Mocha
- Technologies: MongoDB, MySQL, Git, AWS, Heroku

## Projects

- LightningFastAndCutelImageSearch.com - **JavaScript, React, Jeft, Mocha, AWS**  
A unique take on image search, combining multiple image sources in performant experience. The site is open-source, fully tested and has received praise on Reddit. See the [code on GitHub](#).
- Insurance comparison tool - **JavaScript, React, Redux, MongoDB, TailwindCSS**  
Built a tool used by 500 businesses. Contracted with a niche insurance provider to compare small-business insurances. See more details [here](#).
- Data structures in JavaScript - **JavaScript**  
Implemented the most popular data structures like hashtable, heap, queue, stack and others. 100% code coverage on testing. See the [code on GitHub](#).
- Reversible - **JavaScript, React, Redux, Express, MondoDB, Jest, Mocha.**  
A tested, full-stack app with 100% code coverage. See the [code on GitHub](#).

## Experience

Senior financial analyst	Morgan Stanley	2016-2018
--------------------------	----------------	-----------

- Executed on several high-profile projects, resulting in \$15M additional revenue generated.
- Led a financial literacy initiative, hosting trainings across the company, helping people grow their understanding of finance fundamentals.

Finance team lead	Morgan Stanley	2013-2016
-------------------	----------------	-----------

- Responsible for the lending product in New York, leading the team of 5 finance analysts.
- Mentored and coached people on my team to grow, helping 3 of them getting promoted.

## Education

Software Engineering coding bootcamp	2020
--------------------------------------	------

Flatiron School, New York

- Went through a rigorous course that covered the MERN stack.
- I followed engineering best practices, paired with fellow students, and completed a paid client assignment.

BsC, Finance and Accounting	2013
-----------------------------	------

University of New York

## Interests

I'm an avid biker and have cycled across eight states as part of a long-range bike trip.

---

This resume focuses on the right details. It starts with a short summary, giving context about the previous profession of this person. As this person doesn't have full-time working experience as a software engineer, the next section is about languages they are proficient with, and the projects they've built. The emphasis—and bolding—is on the technologies. This is what the recruiters will be most interested in.

After this section, there's a brief experience of the work summary. In this case, there are no software engineering related positions, and the resume keeps it short. There's one sentence about the position and another about some skills that could be relevant for developers. This person led a training initiative, and they mentored and coached people.

The resume closes with education. The software development related education has more details, as that's the more relevant one. If this person had related courses or certifications, listing those here would also be a good idea. Listing non-software-related university degrees is good practice: degrees, even from different fields, are usually worth sharing.

## Career Breaks

You might have gone for many months—or years—without a full-time job, after leaving or being let go. This might be because you did not want to pursue regular employment, or because you could not find a suitable job at a time. Either way, there's a gap on your resume. How do you present this?

While there is no “one size fits all” advice, here are a few good approaches:

- **For a career break that was more than a few years ago**, you don't need to explain it. Hiring managers and recruiters will rarely care for breaks that happened more than 4 or 5 years ago. Even if they do, this would come up during a screening call, and shouldn't make a difference for the resume screening. Don't waste the space explaining it.
- **For a recent career break that does not lead up to the present**, use good judgment on whether you want to add it or not, and how to tell your story. As you have been employed up to now, a gap from earlier might not be as relevant. Still, you might decide to give clarification.
- **For a recent career break leading up to the present, tell a story**. If you have been out of a full-time job, take the lead with the narrative. You'll want the resume to say something similar to what you'd answer to the hiring manager if they'd ask about this break. Here are a few career break examples that are clear to understand:

### On a break

Mar 2020–Aug 2020

- Due to COVID-19, I decided to take a break to spend more time with my family.

### Previous Position

Jan 2019–Mar 2020

- ...

### Travelling & writing

Jan 2020–Present

- I took a longer professional break to travel to Asia, and to write a novel: both things that I had been wanting to do for a long time.

### Previous Position

2016–Jan 2020

- ...

Career breaks are not unique to tech, and as such, advice on how to present these are not unique, either. The article [How to explain the time off during your career break](#) on Monster UK gives some good suggestions on how to word specific situations:

*You went traveling: "I fulfilled my life-long dream of traveling through South America for six months, where I improved my Spanish and gained an understanding of different cultures."*

*You wanted a fresh start: "I chose to take time off work to pursue my interest in painting and had the time to decide on a new career path that really fits my personal goals."*

---

*You were ill: "I took time off work due to a lengthy illness, but the time off allowed me to fully regain fitness for the long-term and I became involved in volunteering at a local charity shop to get me back into the swing of work."*

*You were caring for someone else: "I took time off to care for my brother, who has now made a full recovery."*

## Senior and Above Engineers

The more experienced a software engineer you are, the more likely you will stand out based on your experience and past titles. Depending on the type of companies you are applying to, consider the following.

When you have a senior title, but not more than a few years' work experience, some companies will focus more on the years of experience than your current title itself. If you have been promoted, make this promotion visible from your resume. Be clear about your work's results and impact to give context on the output you have delivered.

With more than 8-10 years of experience, you likely have worked with several different technologies. Your challenge will be to keep your resume relevant to the position you are applying for.

- **Add a summary section** tailored for the position. You are experienced enough that both the recruiter and the hiring manager will be interested in your motivation and how you can help their team.
- **Mention “soft” achievements on top of the business results.** These can include mentoring more junior developers, leading teams, and other activities where you have helped others. Many companies will look for seniors who have a track record of not only doing their work but also helping people around them.
- **Mention not only impact but also your influence.** On top of the business impact, you can mention what kind of impression you made with your work and what teams or organizations you influenced for specific outcomes. The more senior you are, the more you are expected to be able to influence. For example, staff-level engineers at the likes of Facebook, Google, and similar companies are expected to influence multiple teams. Recruiters at these places might be looking for these kinds of track records.
- **Move the Education section to the second page.** You should have more than enough work experience to fill up one page, and education becomes less important at this level. Also, by moving it off the first page, you can reduce age bias. The person reading your resume won't be able to think about your age on the first page—they can only do the math of looking at your graduation date on the second page. And by this time, they have probably decided whether you are qualified, based on your skills.
- **For Education, only keep standout details.** Your education section should be a line or two, with the name of the school, degree, and date. Only keep standout achievements such as summa cum laude, perfect GPA, or a relevant specialization. Other details will be irrelevant for the hiring manager, in this timeframe.
- **Make your earlier work experiences more concise.** The further in the past your work experience, the less relevant it is. You should take up proportionally less space for positions in the past, reducing the information you share.

With plenty of relevant experience, you can also consider sidestepping the direct application channel. Try to find the recruiter and hiring manager on LinkedIn, and directly messaging them, inquiring about applying. For more experienced engineers, the people doing the hiring are more likely to respond. Once you have a point of contact, you can follow up on your application progress more easily.

## Tech Leads

The role of “tech lead” is often advertised as technical team lead, team lead or technical delivery lead. Unlike a senior engineer, these lead roles focus less on technical contribution, and more on delivery and team leadership duties. While an engineering manager might be responsible for several teams, a tech lead focuses on just one team. There can be a few companies that advertise this role as an engineering manager one, in cases when they mean to define a [Tech Lead EM archetype](#) as Patrick Kua describes in the [5 Engineering Manager Archetypes](#) article.

When applying for tech lead positions, follow these pointers as guidance.

- **Showcase how you helped teams that went beyond what senior engineers do.** Hiring managers for these roles will have different problems they need to solve than those recruiting for (senior) engineers. They may feel the team is behind on delivery, or the team may be struggling with high turnover, or they may be delivering poor quality—but they hire a tech lead when the problem isn’t solvable by adding more people to deliver code. Did you speed up delivery on your past team? Improve quality? Repaired stakeholder relationships? Make all of this clear on your resume.
- **Give examples on how you made your team successful.** You will be expected to have a track record of making teams better. Make the evidence easy to find.
- **Talk specifics, and give context.** Recruiters will be interested in the size and make-up of the team (e.g. “2 front-end engineers, 2 back-end engineers and 1 test automation engineer”), the context within which the team operates (“part of a 50 FTE team building...”), and the major technical choices. Make these clear.
- **Outcomes, as well as activities.** Talk about both in your resume. It’s reassuring to know you ran the daily scrums, but did the team deliver on time? If you can, make sure your resume includes proof points for delivery (“managed a team that delivered project x, on time and on budget”), team leadership (“coached and mentored 2 graduates who became senior engineers”), and technical leadership (“worked with the architecture team to deliver essential component x”).

You will probably find fewer tech lead positions advertised compared to senior engineer and engineering manager ones. You might decide to apply for these positions as well. You should create a separate senior engineer and a separate engineering manager resume for these positions, highlighting your work experience relevant for each track.

## Engineering Managers

When applying for engineering manager roles, it is common to expect that you've had an engineering manager, tech lead, or similar position with your previous jobs. Assuming you have this, the below advice can help with your resume.

- **Tell a story.** Engineering managers need to be good storytellers—and your resume should showcase this. Tell the story of how you made teams better, how you achieved great results, and what you'll help the organization to achieve, should you join.
- **Be familiar with the values of the company.** Engineering managers are never hired in a vacuum: they are hired to keep teams in the organization thriving and to take them to the next level. Do your research on what the culture of the organization is. Think about which of these values you already represent. Reflect on them when you talk about your past achievements.
- **Have a strong and relevant summary.** As an engineering manager, your summary is your cover letter. Craft it carefully, tailor it to the position you apply to. Avoid buzzwords and overly generic summaries. Have it represent you well.
- **Role fit is very much a two-way street—don't force it too much.** Getting your next job as a senior engineer would likely be much easier than as an engineering manager. Teams and companies look for a specific fit to fill the manager gap they have. While you might be tempted to edit your resume to make it seem you'd be this fit, think carefully. Your goal should be to only progress to a hiring manager conversation at places where you can play to your strengths. This is why being clear on the environments in which you deliver the best results is so important.
- **Make your achievements not just about the business results, but also about the team.** You've shipped impactful projects, achieved challenging OKRs, and took a cross-team initiative to launch a key project. That's all great. What about your team? How did you support people? Did you help people get promoted? Were you able to hire for diversity? Did you have low attrition? Were you a coach or mentor to people outside your team? People are at the heart of engineering management—paint the full picture of how you did here.
- **Get inspired by others' LinkedIn.** Writing a good engineering resume is hard, as there are few examples and resources. Get more inspiration by browsing the LinkedIn profiles of fellow engineering managers. See how they talk about their work, impact, and results.
- **Reach out for a chat with the hiring manager.** As an engineering manager, it's easier to have an introductory chat with the recruiter or hiring manager, compared to most developer roles. Consider reaching out this way before directly applying.

[Neville Kuyt](#), who has been leading teams for over two decades, shares his additional advice:

- **Have a summary that positions you for a specific niche.** Back these claims up in your work experience section. Many summaries are full of buzzwords, and they could be so generic that they could apply to anyone. You'll want to aim for the opposite.
- **Talk about the type of challenges you are a perfect fit to solve.** Take a step back and think about what types of problems made you achieve your best results. Capture these challenges in your summary. Then back it up with evidence in your work history.
- **Target the right types of companies** in your applications. Are you someone who thrives in a large organization, driving alignment? Someone who does well in early-stage startups, where being hands-on is a must? Apply for the positions where you would—or have—done well.

---

**Get more feedback** and look for other ways to do so. If you have engineering manager peers or friends, ask for their input for your resume. As an engineering manager, you can get resume feedback advice from fellow managers in the #vet-my-resume channel in [Rands Leadership Slack](#). If you are not a member, you'll have to join this community first. Please, read the Code of Conduct, follow it, and contribute back to this community. I suggest joining the community well before you'd need feedback on your resume and participating in other discussions.

# Chapter 9: Exercises to Polish Your Resume

Once you have created your “master” resume, here are a few exercises to tweak it

## Write Two Different Resumes

The people who write great resumes have written several of them. The people whose resumes are not fantastic usually keep tweaking and tweaking the same one. Do this challenge to have a significantly better resume.

First, write a resume using your preferred template. You have probably already done this. Then, start fresh, using a completely different template. It can be a template you’re not in love with; but it should be different. Put away your existing resume; you’re not allowed to cheat.

Write a brand new resume, aiming to write a completely different one. Don’t repeat the achievements that you have written on your previous one; come up with different ones. Change the structure. Be bold with using colors. Come up with something very different.

Now, compare the two. What are things you could use from this second one that seem pretty good? With two very different resumes, come up with a third, where you take the things that you like from both resumes. You’ll not only have a better resume—but you now have a lot more details to choose from when you customize your resume for the specific job listings.

## Find Out the Impact of Your Past Projects

Many resumes list out responsibilities, but they don’t mention impact or specifics. They might say “Built and shipped the website rewrite for MyCompany”, but they don’t talk about how many people viewed the website, how much incremental revenue it generated, by what percentage did the loading times improve, by what percentage did the codebase shrink, and so on. In many cases, this is because the person writing the resume is also unaware of the actual impact.

So find it out. For your current position, ask around on your team and through your manager. What is the expected impact of the project? Or what was the impact of the launch? For past positions, reach out to your old team or previous managers and ask about the impact. They might have better pointers on what this was.

Finally, if you don’t have an exact impact, estimate to a good approximation. Can you give a rough percentage on the improvement made? A rough estimate on from what number to what number things changed? With impact, you don’t need to be exact. “Close enough” is better than nothing.

## Do a Grammar Check, Not Just a Spellcheck

Your resume should read naturally: no typos, and clear grammar. The sentences should be short and easy to read. For grammar check, you can use the free version of [Grammarly spell checker](#). For easy to read text and sentences, use the free [Hemingway Editor](#). Don't only rely on these tools, though: re-read, re-read, re-read your resume.

The screenshot shows the Hemingway Editor interface. At the top, there's a toolbar with buttons for Bold, Italic, H1, H2, H3, Quote, Bullets, Numbers, Link, Write (which is selected), and Edit. Below the toolbar, the text "Hemingway App makes your writing bold and clear." is displayed. This text contains several highlighted errors: "lengthy, complex sentences and common errors" (yellow), "shorten or split it" (yellow), "If you see a red highlight, your sentence is so dense and complicated that your readers will get lost trying to follow its meandering, splitting logic" (red), and "try editing this sentence to remove the red." (red). To the right of the text, there's a readability analysis section with the title "Hemingway Editor". It shows a "Grade 6" rating, "Good" readability, and 133 words. Below this, there are five colored boxes with feedback: a blue box for adverbs, a green box for passive voice, a purple box for simpler alternatives, a yellow box for hard-to-read sentences, and a red box for very hard-to-read sentences.

Hemingway App makes your writing bold and clear.

The app highlights lengthy, complex sentences and common errors; if you see a yellow sentence, shorten or split it. If you see a red highlight, your sentence is so dense and complicated that your readers will get lost trying to follow its meandering, splitting logic — try editing this sentence to remove the red.

You can utilize a shorter word in place of a purple one. Mouse over them for hints.

Adverbs and weakening phrases are helpfully shown in blue. Get rid of them and pick words with force, perhaps.

Phrases in green have been marked to show passive voice.

You can format your *text* with the toolbar.

Paste in something you're working on and edit away. Or, click the Write button and compose something new.

2 adverbs, meeting the goal of 2 or fewer.

1 use of passive voice, meeting the goal of 2 or fewer.

1 phrase has a simpler alternative.

1 of 11 sentences is hard to read.

1 of 11 sentences is very hard to read.

*The Hemingway Editor—an excellent tool to make your resume easy to read.*

## Ask for Friends or Family to Proofread

As you write your resume, you will develop tunnel vision and ignore obvious inconsistencies or mistakes. Get a fresh pair of eyes to read through your resume, someone who can catch things that don't look or sound right.

Ask a friend or a family member to read through your resume thoroughly and give you feedback. You'll get feedback on things—grammatical issues, typos, logical mistakes—that you would have otherwise missed.

## Get Feedback on the Internet

There are several forums where you can get feedback on your tech resume—for free. As with most things free, don't get your hopes up too high. These are forums where members do a favor for giving any feedback. My suggestion is to join these communities well ahead of time, read feedback on other resumes, and treat the advice with a grain of salt.

**Remove personal information from resumes** you submit to forums like this, as the resume might be circulated well beyond your control.

- Reddit has regular [resume advice threads on CS Career questions](#). This forum is meant for new grads.
- Reddit has a general [resume review sub](#) where you can get feedback on your CV/resume.
- If you are an engineering manager, you can get resume feedback [within the Rands Slack channel](#).
- Search for resume review offers from fellow members of the tech community. Dev.to had a [resume review thread](#) and people occasionally offer to give feedback on Twitter and on LinkedIn.

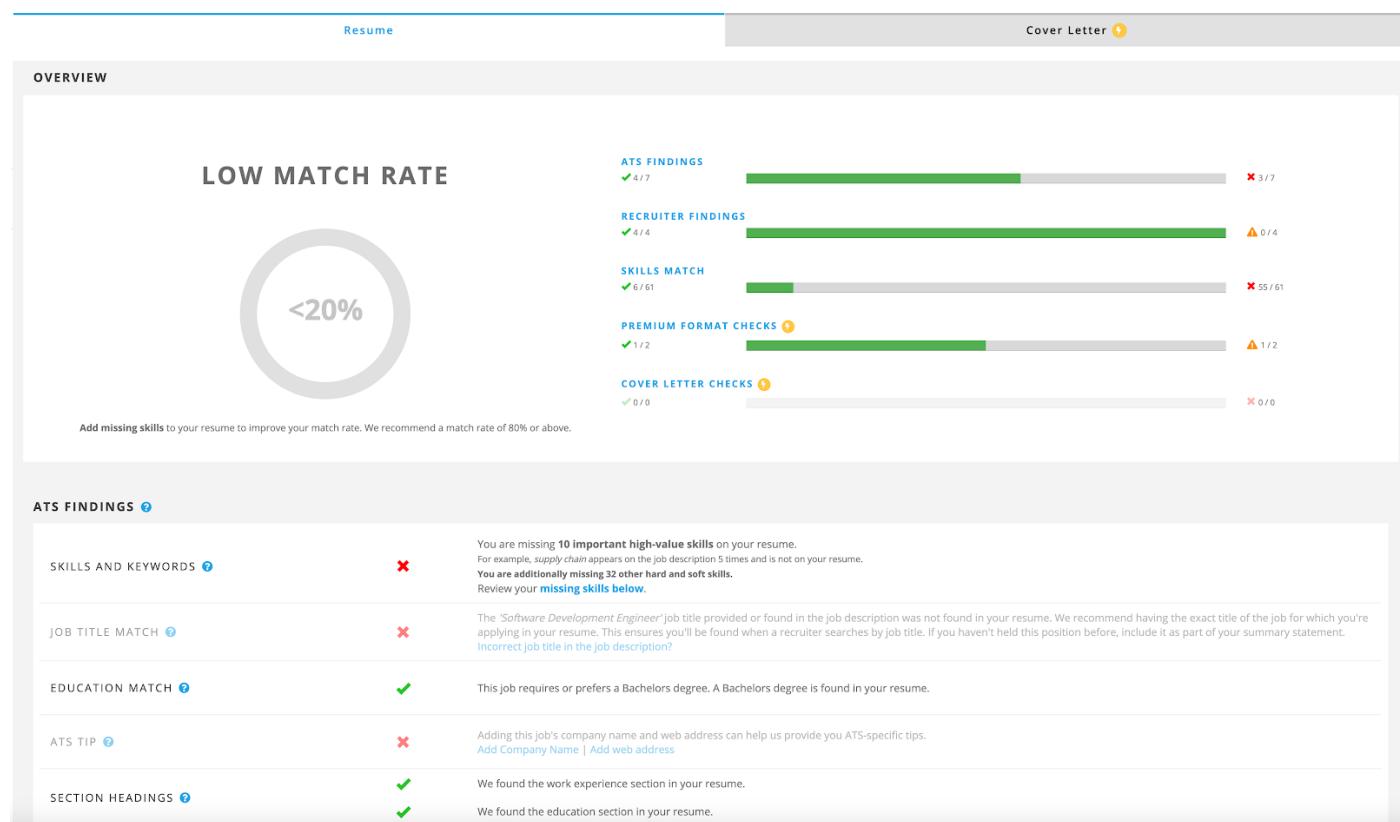
## Keyword Check for That Position

When submitting a resume for a specific position, if your resume mentions things that are considered keywords in a job, you'll be more likely to stand out. For example, for a Java backend position, you'll want Java and backend mentioned—assuming you have experience with both.

A possible way to do this review is for you to compare the job description and your resume. There are services that offer a more detailed scan. Most of this scan is guesswork, but it can give you a sense of how much words overlap your resume and the job posting has.

**I hesitate to wholeheartedly recommend ATS scan services**, as most of these scans are based on ATS fallacies, and optimize to make you pay up for ATS optimization, that will not be particularly helpful. See the [ATS Myths Busted section](#) on more details on why most of the claims made by these sites do not apply to tech positions.

Still, doing a quick check with JobScan ATS scan can give you some pointers on the overlap between your resume and the position. Keep in mind that most of the ATS scores are guesswork, and ATSSes will not reject your resume. And I do *not* recommend upgrading to JobScan Premium for tech roles, as much as the site tries to upsell to do this. For example, you will always see a lower than 100% "match rate" in this scan that incentivizes opening your wallet and pay for advice that will likely be irrelevant with tech roles.



The JobScan ATS Check. Much of it won't apply to how tech companies use ATSSes. It could give you ideas on how to tailor your resume for the job description, though.

---

## Recap: Actions to Improve Your Resume

In this chapter, we've covered additional exercises to make your resume better. Consider doing multiple or all of these to make your resume even better.

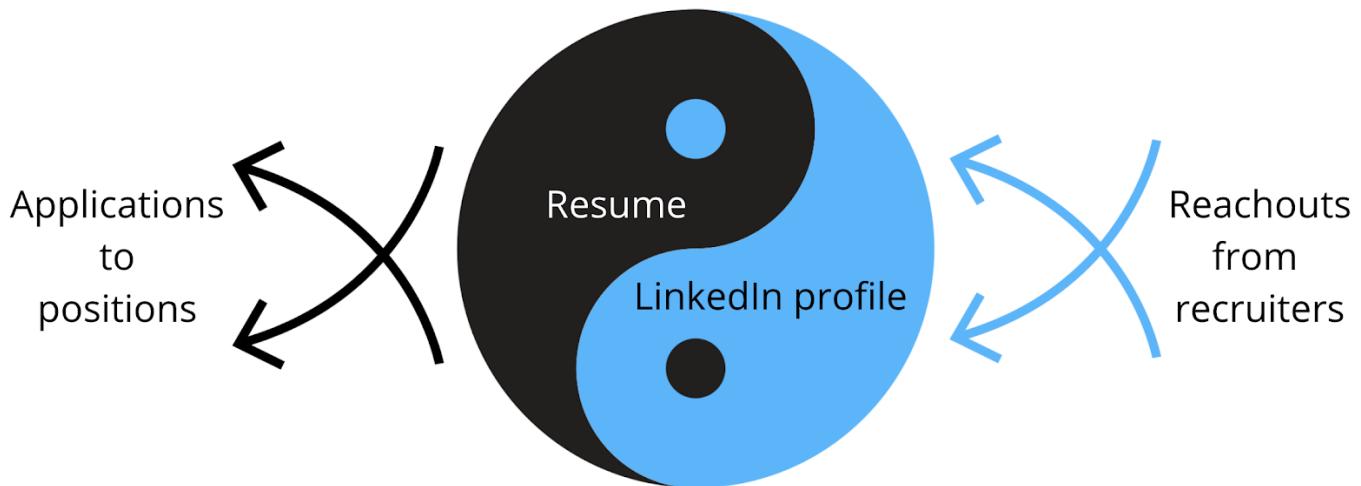
1. **Write two very different resumes**, then come up with a third one, combining the best parts from each one.
2. **Find out the impact of your past projects and add them to your resume**. Reach out to past colleagues, or do a rough estimation. A close enough estimation is better than no specifics.
3. **Do a grammar check, not just a spell check**. Use tools like [Grammarly](#) or the [Hemingway Editor](#) to do so.
4. **Ask friends and family for proofreading**. You'll get a fresh pair of eyes, and valuable feedback on things that you might have missed.
5. **Seek out feedback on the internet from community forums**. Don't forget to anonymize your resume first. Treat this feedback with a grain of salt, though.
6. **Do a keyword check for your resume**. Use either an automated tool or just "pretend" to be the screening software, looking for keywords that you pick up from the job description.

# Chapter 10: Beyond the Resume

Your resume is not the only thing that can get you to a recruiter call: cover letters, your LinkedIn profile and a few others are worth paying attention to.

## LinkedIn Profile

Resumes and LinkedIn profiles are like the yin and yang. You directly apply with a resume, and opportunities can come and find you with your LinkedIn profile. It's smart to invest equally in both.



The good news is, once you have spent time polishing one, you can reuse much of the content for the other. There are a few differences that are worth keeping in mind with LinkedIn, though.

Most of these differences come from how recruiters use LinkedIn most of the time: by looking at the LinkedIn search view. They set out to find candidates who match a specific skillset, in a certain location, and might narrow down to other filters: companies, schools, and others. Here how recruiters will see your profile in the LinkedIn recruiter search view:

The screenshot shows the LinkedIn search interface. On the left, there's a sidebar with filters for Job titles, Locations, Skills, Companies, and Schools. Arrows point from the 'Skills' and 'Companies' sections to the main search results area. The main area displays search results for 'Finley Williams' and 'Kellan Burke'. The results are annotated with numbered labels pointing to specific details:

- 1. Headline / current title, location, industry**
- 2. Current title, date**
- 3. Past position titles & dates**
- 4. Education: school names**
- 5. Boolean search for skills / technologies**
- 6. Other filters (titles, locations, companies etc)**

The LinkedIn recruiter search view, and the key details recruiters immediately glance at.

To optimize your "search card", consider following these steps:

- **Set a headline to represent what you want to be found for.** This line is the first thing a recruiter will see, and it should summarize the message you'd like to convey. Why should someone click on your profile? If you do not set anything to your headline, your current position will appear here.
- **Tweak your current position** to describe your role better. LinkedIn is not your resume, and your current position does not have to be your official title, as per your work contract. For example, if you are leading a team as a frontend developer, you could have your current title on LinkedIn say "Frontend lead" or "Frontend team lead", even if your official title at work is "Software Developer". Similarly, you can change your current title to say "Backend engineer specializing in Go" if this is both what you do and how you'd like recruiters searching on LinkedIn to find you.
- **Mention keywords in your summary and job descriptions** that recruiters could search for. These include technologies, frameworks, and engineering practices.
- **Use a professional-looking photo** that makes you look good. A professional photo is one where your face can be seen clearly, with no one else in the picture. Note that you don't need to have any photo on your profile, especially as photos can encourage biases. Also, no photo is better than a poor photo, where you look grumpy or unfriendly.
- **Have a summary that sells you.** Unlike with your resume, where recruiters and hiring managers barely look at your summary, on LinkedIn they are more likely to do so. As a hiring manager, I read the summary section and the last one or two job descriptions. So go for a longer summary, up to two paragraphs, where you describe the value you can bring and your motivation.
- **It's fine to be more verbose on your LinkedIn than you would on your resume.** The nice thing about LinkedIn is you don't have to fit everything on one page. In fact, with more content, you could rank higher in search results than if you kept things concise.

- 
- **Omit work experience that doesn't support your professional career story.** Similar to your resume, your LinkedIn profile should tell a good story of your professional side. Remove positions that don't strengthen this image. For example, if you did a part-time job years back that has nothing to do with your current profession, it probably will not add anything to your career. Don't have this kind of noise.

## How recruiters do Boolean searches

To optimize your LinkedIn profile for recruiter searches, let's take a look behind the scenes on how recruiters search for candidates. They use the technique called Boolean search, something that should ring familiar to most developers. Using a series of AND, OR, and NOT operators, they filter details. Here are a couple of possible searches recruiters might use:

- Software developer: "software developer" OR "software engineer"
- Fullstack developer with React.js and Python experience:
  - "React.js" AND "Python" for a simple search
  - ("developer" OR "engineer" NOT "manager") AND React.js" AND "Python" for a more sophisticated search
- Frontend developer with React or similar experience: ("frontend developer" OR "front-end developer") AND ("javascript" OR "typescript") AND ("react" OR "reactjs" OR "vue" OR "vuejs" OR "angular" OR "angularjs")

Recruiters will often Boolean search for keywords that are in the job description that the hiring manager gives them. So taking this example from a job description, highlighting possible keywords:

- Computer Science fundamentals in data structures, algorithm design, problem solving, and complexity analysis
- Proficiency in at least one modern programming language such as C, C++, Java, or Perl

A recruiter could create these Boolean searches:

- *Focus on languages:* "C" OR "C++" OR "Java" OR "Perl"
- *CS fundamentals and languages:* ("data structures" OR "algorithm" OR "algorithms" OR "design") AND ("C" OR "C++" OR "Java" OR "Perl" OR "object oriented")

If your profile contains these phrases, you'll appear in the results. The closer the match—for example, when matching in the headline—and the closer you are in your network with the recruiter, the higher you will rank on the search results.

## Optimizing LinkedIn when job searching

When you are either actively job searching or are getting ready to start, consider the following:

- **Signaling you're open for opportunities: don't do this in your headline.** Do not add "Looking for opportunities", "Open for a new role" or something similar in your Headline. This won't add much value, and it will take up valuable space that you could use to sell yourself and highlight your skills. Instead, activate the Open to Job Opportunities feature in your LinkedIn profile:

### Let recruiters know you're open to opportunities

Share that you're open and appear in recruiter searches matching your career interests

We take steps not to show your current company that you're open, but can't guarantee complete privacy. [Learn more](#)

Yes  Saved

- **Always have a current position listed,** even if you don't currently have a job. LinkedIn ranks your profile lower when you do not have a current position. [Kathy Bernard](#), who specializes in optimizing LinkedIn profiles, advises to always fill out something as content here—even if you don't have a job right now. As a developer, you can tailor this current position to showcase the technologies and languages you are an expert in. For a few more pointers, see her article: [Unemployed? Create a "current" role on LinkedIn](#).
- **Add more headline details.** Consider using the "|" separator character to add more information to your headline, such as key skills or technologies you'd like to be found for. For example, a headline could be:  
*Full-stack dev with 5 years experience | Go, Java, JavaScript, React, Swing | Distributed systems, developer tooling & fluid user interfaces.*  
A tailored headline will rank your profile higher with keyword searches and grab the recruiter's attention better.
- **Connect with more people on LinkedIn, including recruiters.** The closer you are connections-wise to the recruiter doing the search, the higher your profile will rank. As you start your job search, do consider adding current and former colleagues and recruiters who have previously reached out to you.
- **Consider updating your location when you have a definite destination in mind to move to.** Location is often a filter that recruiters use. If you are ready to move to a city or a country, you can update your location, so you show up as a local candidate. Use good judgment, especially for cases when you'd need a visa to move. Updating a location will most likely result in more inbound messages from local companies. Once you have a reachout, you can confirm if the company sponsors visas before investing more time in the process.
- **Use Boolean search when searching for jobs.** It's not just recruiters who can do Boolean searches: on job sites like LinkedIn, you can do so as well. You can use AND, OR and NOT for phrases, and parenthetical search. For example, you could use the following search string to look for non-senior frontend jobs, for companies not building on top of the Laravel framework:  
`(frontend OR javascript OR react) AND NOT (senior or laravel)`  
Narrow your search to fewer opportunities that are a better match for what you are looking for, or your skillset. This will help you apply in a more focused way.

## GitHub

There is no expectation of having GitHub on your resume, regardless of how many resumes and resume templates you might see these included in. As a hiring manager, I never explicitly look for either. It is a nice bonus to see it, though, and as a hiring manager, I frequently click through to check it out.

Other hiring managers confirmed the same: they are more likely to click through and check out these profiles after the first scan. Recruiters are somewhat less likely to do so: though the more technical the recruiter, the more inclined they are to take a look. Should you make it to the onsite, developers who would be interviewing you often click through and bring up the projects as talking points on the interview. Which brings us to the most important rule of linking to GitHub.

**Only include a GitHub link if there is something you are proud to show off** on these sites. Do not link to an empty Github profile or one that has poor code visible.

Clean up your GitHub profile to describe the work that people will find there if you do link it. Add proper READMEs to your main projects that give context on what the project does, highlighting relevant code, setup, or configuration.

As a hiring manager, here is what I look for when browsing through a GitHub profile:

- **Context:** what side projects has this person published? What do they do? Do the pinned projects have a clear description on the profile page? When clicking into the project, is there a clear README that describes the project: how to run it and how to test it?
- **Code quality:** how is this? Does it follow best practices? Is naming decent, the code easy to read and clean?
- **Testing:** does the project have unit or integration tests? If they do, I know this person is aware of testing and has at least played around with it.
- **Contributing to larger projects:** what larger open source projects have they contributed to? I sometimes look at pull requests, and the comments on the pull request, if it's easy to find.
- **Standout repositories:** if there is a repository that has a large number of stars, and lots of activity, how does this developer handle pull requests, and how do they respond to issues? This can give some good, positive indication on their collaboration skills, similar to contributions to larger projects.
- Things I don't pay attention to are contributions in the last year, followers, numbers of repositories, and other statistics that have little to do with code quality.

To optimize your GitHub profile that you are linking on your resume, follow these few steps:

- **Create a README for your profile** and summarize things that a hiring manager should know about you, pointing to standout projects. To create a personal README, create a repo with the same name as your username, and initialize it with a README:

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

### Repository template

Start your repository with a template repository's contents.

No template ▾

Owner \*



gergelyorosz ▾

Repository name \*

gergelyorosz ✓



You found a secret! [gergelyorosz/gergelyorosz](#) is a ✨special✨ repository that you can use to add a README.md to your GitHub profile. Make sure it's public and initialize it with a README to get started.

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.

Then take inspiration from other developer READMEs in creating yours from lists like:

- [Awesome Github Profile READMEs](#) by [Abhishek Naidu](#)
- [Awesome Profile README templates](#) by [Kautuk Kundan](#)
- [Awesome Readme Templates](#) by [Elango Sundar](#)

- **Pin the “best” repositories on your profile**—up to six of them. You can control which repositories are shown on your profile. Hiring managers will most likely only click on a few of these, so make them count.

**Gergely Orosz**  
gergelyorosz

Engineering manager & building things @ Uber, Skyscanner, Skype & JP Morgan alumni.

[Edit profile](#)

**Popular repositories**

- swift-objective-c-mixed-project**  
Creating a mixed Swift 2.0 & Objective C project  
Objective-C 46 stars, 9 forks
- GhostSocialCasper**  
An extension to the Ghost Casper theme, adding a sidebar and social navigation links  
CSS 15 stars, 3 forks
- SwiftAlgorithmsAndDataStructures**  
Swift implementation of popular algorithms & data structures using Swift  
Swift 4 stars
- BlogSkins**  
Ghost skins for different blogs that I run  
CSS 1 star

**Customize your pins and link your "best" repositories**

- **Fill out descriptions** for each of the pinned projects in the repo's description field. The description should make it clear what the project does and why it is interesting. You can make the descriptions fun: your goal is for the person to check out the repository.
- **Have great READMEs for your pinned projects.** At the very least, the README should describe what the project does and how to use it. Better READMEs have code examples, show how to configure, or list further improvements. Browse READMEs of popular projects and consider improving your README to be more similar to them. You can use the [README score](#) tool to get feedback on areas you could further improve.

## Technical Blogs

Do you know what percentage of candidates write technical blogs? Only a fraction of all resumes I've seen. So if you happen to write one, you'll already have something to stand out with—assuming you add it to your resume. However, don't *just* add your technical blog to your resume. Similar to your GitHub, be deliberate about what the one or two articles you'd want a hiring manager to read.

### Before and after: technical blogs

Take these two resume examples. The first example is from a resume where the person is applying for a frontend position that is asking for some experience with Angular.

#### Before:

##### Projects

- I write about technical concepts that I learn about or find useful for the developer ecosystem on the [Dev.to](#) blogging platform.

#### After:

##### Projects

- I write a technical blog. Popular articles include a piece [on Angular Material](#) and one on [The Angular roadmap: looking back, and looking ahead](#).

In the "before" version, when the hiring manager clicks on the link, they are taken to the linked Dev.to blog. On this page, the last two articles are non-technical posts, and buried many pages down are two solid articles about Angular.

The "after" version does a much better job guiding the hiring manager to the post you want them to read. And they don't even need to click through to think, "okay, so this person doesn't only say they know Angular: they've proven it, by blogging about it."

**Add links to the most relevant one or two technical articles** that you have authored in your resume, describing what they are about, similar to the above example. Doing this over "just" listing your blog will let you control more what you'd like to show. You'll also guide your reader better, having them start reading some of your best or most relevant content, should they click through.

## StackOverflow, Twitter, Instagram, Quora and Other Social Sites

StackOverflow is the only other piece of content that I'd suggest to include: but only if you are active on the site, and your profile supports the story you want to tell. For example, if you are a JavaScript expert and have many JavaScript-related badges, this can support your credibility as someone who knows this technology. As a hiring manager, I click through far less to StackOverflow links in the resume than I do to GitHub, though.

For other social media like Twitter, Instagram, Quora and others: I discourage linking to these on your resume. These mediums do a poor job conveying professional value to the hiring manager and the recruiter.

## Cover Letters

Big tech companies rarely ask for cover letters—they will only look at your resume. Small and mid-sized companies allow attaching these, but very few explicitly ask for it. Most tech recruiters I've talked with don't ask for it, and even if sent separately, rarely read them.

Cover letters in tech matter at companies where the hiring manager does the resume screening. These are typically smaller companies, often startups, which see fewer applicants for roles. As such, I suggest prioritizing writing cover letters to smaller companies only, or in cases where you know you can reach the hiring manager directly.

Also, your tailored resume for the position should already be a cover letter. If you follow the advice in the section [Write a Resume for That Job](#), the cover letter will be less relevant. You never know if the cover letter will be read or forwarded, but as long as you progress in the hiring process, your resume will always be.

### ***From the inside out: what matters for a cover letter?***

While larger tech companies often don't ask for cover letters due to the large number of applicants, this is not true for smaller ones and for startups. In these environments, cover letters can be a real advantage. Here's what [Monica Lent](#) has to say, who reviewed thousands of CVs while hiring in Berlin for SumUp, a high-growth startup. The advice as part of her [seven software developer resume tips](#).

*"Even if you think you have a highly relevant CV, you should still consider writing a cover letter. It doesn't have to be long, but it can only increase your chances of passing the CV screening phase.*

*Companies you're applying for want to feel a little bit special. Like if they were to offer you a job, there's a good chance you'd actually accept it. You will stand out if you have a coherent cover letter that demonstrates you've read the job ad and the website. It doesn't have to be long, either. In fact, it shouldn't be long, because ain't nobody got time to read a super long cover letter. And don't forget to spell check.*

Here are the basic elements that a cover letter for a developer role should have:

1. Show off your good communication skills
2. Demonstrate an understanding of the role
3. Briefly explain your qualifications and why you think you're a good fit
4. Mention the company name
5. Demonstrate that you read the company's website
6. Attach the cover as a PDF. It's just easier to read than pretty much any text format that could be typed into a web interface. Again, optimizing for ease of reading.

*Note that some companies may not find the cover letter useful or even read them. This can vary based on the role – for a typical dev job, perhaps not. But for highly competitive junior roles or very senior roles, your motivation is relevant. In the end, a cover letter will never hurt you, and has the possibility to help you stand out."*

**Tailor your cover letter style** to the company and country that you are applying for. For a young startup in the US, a more casual and outspoken style might work better. For a large company in the UK or Australia, a more reserved approach could work better. For inspiration, see the following cover letter examples:

- [A developer cover letter](#) by principal data engineer Nick Larsen, published on the Stack Overflow blog.
- [A punchy junior developer cover letter](#) case study by engineer Lou Bichard, founder of [Dev Coach](#).
- [A non-traditional cover letter](#) by engineer Santiago Valdarrama. A cover letter like this could work better in the US and for less traditional companies and hiring managers.

## Personal cover letters over templated ones

When you decide to write a cover letter, an immediate strategy is to find a good template, then use that. If you search for cover letter templates, they all look somewhat similar, something like this one:

Dear {X},

I am interested in the {position} advertised on {web site}. With my strong background in {relevant skills} and {number of} years of experience, I am confident I can {mention a problem you can help solve}.

Highlights of my experience include:

- {Requirement for the position #1}: {proof that you've done this}
- {Requirement for the position #2}: {proof that you've done this}
- {Requirement for the position #3}: {proof that you've done this}

I would appreciate the opportunity to discuss this position further.

All the best,

{Name}

Templated cover letters will frequently get ignored in tech. They don't give much more information than your resume. In the example above, almost everything listed in the cover letter is probably part of the CV the person submitted.

**Instead of a template, consider spending more time to write a highly personalized cover letter for that position and that company.** Do this only for cases where there's a chance the person would read this. So do this when reaching out directly to the hiring manager, or applying for smaller companies.

Consider this highly personalized cover letter, shared by software engineer [Santiago Valdarrama](#), tailored for a US-based startup:

Hi Acme! 🙌

My name is **John Doe**. I live in Ohio, and I've been building software for a long time.

Let's jump into it: **we are perfect for each other.**

You are opinionated, bold, and outspoken. You put yourself out there, unafraid of the naysayers. You do what you feel is right and take bets. You show everyone how you are different and why being different is better.

I have a lot in common with you, but I also don't. I have incredible respect for your ideas, but I can't wait to challenge them. You are that future promising amazing work, excitement, endless opportunities to debate, and a lot of unknown.

### **I am ready for this challenge.**

You are looking for help, experience, curiosity, and dedication. You want somebody that you can learn from and grow with you. Let me make my case by contrasting **who I am** and **who I am not**:

- I'm really good at taking on hard problems and bringing back solutions, entirely autonomously. I have excellent communication skills, love to write, learn new things, and share it with others. I love challenges. Going uphill is where I do my best work.
- I'm not good at repetitive tasks or working on projects where somebody else figured out all the fun pieces. I don't do well working on teams that don't care.

Building software is my passion, and on every project, I always give my absolute all.

### **Let's do this!**

I've been working at my current job for the last five years, and I love it, but **I can't pass this opportunity**. This feels right. You feel right.

You won't find more resilience and commitment to help you move forward. I won't find a better opportunity to do what I love. I'll be your best man, and you'll be the platform that'll allow me to make a difference. We are going to be better together. I know that.

When I look at your page, you are clearly missing me.

Let's talk!

John.

As a hiring manager working at an outspoken company that challenges the status quo, and takes on bold bets, receiving an inbound cover letter like this would immediately want to learn more about this person.

### **Here's what this cover letter gets right:**

- **It talks about the company first, the applicant second.** The letter shows the person understands the company values they are applying for. The letter shows they have researched, followed, and know the company, and identify with its mission.

- **It's sincere and passionate.** This letter was not copy-pasted with the usual phrases. A person spent a good amount of time writing down their thoughts. It articulates both why this person is excited about the company and what they have to offer.
- **It's well-written, with high-energy coming across.** The cover letter has been edited to be easy to read, bolding added to the right places. The writing makes their excitement come across clearly. Reading it, I can feel it coming across! While many cover letters sound boring, this one stands out due to its energy.
- **It's bold and concise.** "Let me make my case by contrasting who I am and who I am not" is a bold statement. The cover letter is concise in that the applicant only shares two opposite descriptions of themselves. Still, they give an excellent glance at why they could fit well into the team.

This cover letter also breaks some unwritten rules that many hiring managers are used to.

- **It does come across as over the top** in its style. The style is also what grabs attention. Some hiring managers might decide that it's a bit too much, and too pushy - especially if the company is a more conservative one. From the sounds of it, the application is for a startup, where this kind of enthusiasm could work well.
- **It portrays an overly enthusiastic, somewhat junior applicant.** While this could be fine in many cases - especially if this is your experience - this approach would work poorly if you're applying for a staff or principal engineering position. Do be mindful of how you want yourself portrayed when you write your cover letter.
- **The tone might not be suitable outside the US.** Different regions call for different tones, even to stand out. This letter would end up in the bin for a more traditional company in the UK or Australia, where a professional tone is an unwritten expectation.

Use good judgment when writing a cover letter. But don't forget that your goal is to get your foot in the door - assuming your skills are a good match. For the smaller companies that would read a cover letter, a tactful and personalized approach could make the difference.

---

## Recap: Actions to Improve Your Application Beyond the Resume

In this chapter, we went through why LinkedIn, GitHub, StackOverflow, technical blogs and cover letters can also play a part in your application process. Of all of these, LinkedIn, GitHub and technical blogs are usually the most relevant ones for software developers. To increase the chances of a recruiter call, do the following exercises:

- **Update your LinkedIn based on what you want recruiters to find you for**, not necessarily what you are working on right now. Update your headline accordingly, and tweak your positions to better describe the story you want to tell. Mention keywords in your description that you want to be found for.
- **Make sure your LinkedIn is professional**. If you choose to go with a photo, have a good one. Omit work experience that does not add to your career story.
- **If you are looking for a job**, change your settings on LinkedIn to indicate this. Have a current position added even if you don't currently have a job, update your position based on where you are looking and grow your network on LinkedIn.
- **Make your GitHub account sell you** by handpicking projects you want to display on the landing page, and ensuring these projects are high quality. All projects should have good READMEs and aim to showcase code that you are proud of.
- **Call out specific blog articles on your resume**, rather than just linking to your blog. Mention articles that are relevant for the position you are applying for, or ones that stand out for one reason or another.
- **If writing a cover letter, tailor it to the position**. Be brief, but clear about why you would be a good fit for this role and mention what skills you bring, with examples

---

## Part 3: EXAMPLES AND INSPIRATION

We've covered everything there is to know about writing a good developer resume. In Part 1, we went through the context on why resumes are important and how the hiring process works at most companies. Part 2 gave actionable advice on how to write a good resume, with examples on how to improve parts of a developer resume.

What you have not seen is something to get started with: templates or example resumes. This section is where we cover these.

[\*\*Chapter 11: Good Resume Templates\*\*](#) covers the principles on what makes for templates that are considered good and easy to read.

[\*\*Chapter 12: Resume Templates\*\*](#) goes through the three resume templates that come with the book. We also analyze another eight resume templates that are popular with software developers, going through the good and bad characteristics of them.

[\*\*Chapter 13: Resume Improvements and Examples\*\*](#) takes real resumes from developers who were looking for a job, analyzes them and shows improved versions people crafted after taking advice from this book. Personal details for all the resumes changed to anonymize the resume.

[\*\*Chapter 14: Advice for Hiring Managers on Running a Good Screening Process\*\*](#) is for any current or future hiring manager—even if you are not one today, one day this could be you! We'll summarize ways to build a better resume screening process, to be inclusive of people whose resume would not represent them well.

---

## **Chapter 11: Good Resume Template Principles**

Before choosing a resume template, it's important to understand what principles make for a good template. A good template is one that will help with the goal that you had writing a resume: conveying the information that you intended to communicate in a way that the recruiter or hiring manager can quickly understand.

In a world where a recruiter receives one application to review per day, resume templates wouldn't matter. In fact, most of this guide would be of little use. The recruiter would take their time, read the whole resume, and probably still give you a call to find out if there is any way you'd be qualified for the position. I mean, you are one of the very few applicants!

However, in reality, recruiters and hiring managers often have dozens or hundreds of resumes to go through. They become really efficient at doing so. If a resume proves to be too much work to find key pieces of information, then it's down to luck if they will take this time, or just move to the next one.

Good resume templates are made tailored for recruiters and hiring managers—not for job seekers. It makes the job of these recruiters and hiring managers in finding key pieces of information as effortless as possible.

## From the inside out: why popular resume template sites don't necessarily work for software developers

There are several popular resume template sites that promise to build you a "stunning" or "standout" resume for each industry: whether you are in finance, are a freelancer, in legal, and dozens of other categories. These resumes are often eye-catching, pretty, and give you ideas you've not thought of before. Visualizing how you spend your time? Listing what you are proud of? A novel design that is beautiful and unusual?

The catch is that these sites optimize to convert *you*, the job seeker, to pay them money. They help create resumes that *you* will feel good about. As a candidate, you will feel that you have represented your standout experience with a standout resume. They also guide you in writing down your experience in a structured way—but they do not do this targeted for developers. Take a look at one of the popular resume template sites, EnhanCV, and their value proposition:

The screenshot shows the EnhanCV resume builder interface. On the left, a sidebar lists seven features with corresponding icons:

- Memorable Design
- No learning curve
- Content Analyzer
- Improve with feedback
- Professional, yet personal
- Tailor your application

On the right, a sample resume is displayed for "HANNAH MILLS". The resume includes:

- Profile picture and bio: "I solve problems and help people overcome obstacles."
- Contact information: "+1 703 340 6006" and "hannah.mills@gmail.com".
- Experience section listing roles like "Unit Director" and "Internal Project Manager" with bullet points.
- Industry Expertise section with a horizontal bar chart.
- Tech Stack section listing tools like "Zoho Sprints", "UserVoice", "Intercom", "VWO", and "Taboola".
- Life Philosophy section with the quote "Impossible is potential. Impossible is temporary. Impossible is nothing." by Muhammad Ali.
- A "Most Proud Of" section with three items: "Record Project Delivery", "Mentoring", and "Taking On Challenges".
- A "My Time" section with a circular chart and a list of activities.

*The EnhanCV value proposition. This kind of flashy template is a poor choice for software engineers or engineering managers.*

### These "standout" resume templates are usually a poor choice for software developers.

Even for sites that have a software engineer template, this template was more likely put together by a designer than a technical recruiter. The resumes are usually harder to read for recruiters and hiring managers. As a hiring manager, I find they incentivize sharing fewer specifics and more buzzwords. If going for a resume template or resume builder, aim to go with sites that avoid flashy design, and have a track record of success for software developers.

## Good Resume Template Layout and Principles

A good resume template is easy to read. They are easy to read because they follow common conventions that people are used to. Things that people are used to when reading nonfiction are these:

- **Top to bottom reading.** We start at the top and make our way downwards. This is how we read books and magazines.
- **Important things first,** less important things towards the end. Good nonfiction reading materials start with an introduction that sets the context and often explains the chapters to come. For a resume, in a somewhat similar fashion, the most relevant things should be towards the top, with the less relevant things on the later pages.
- **Strategic use of colors and bolding.** Bolding can draw attention. **Colors can also pull your attention.** Use both of these in a strategic way to guide the reader on things that you'd like to stand out.

### From the inside out: what direction do recruiters and hiring managers read resumes?

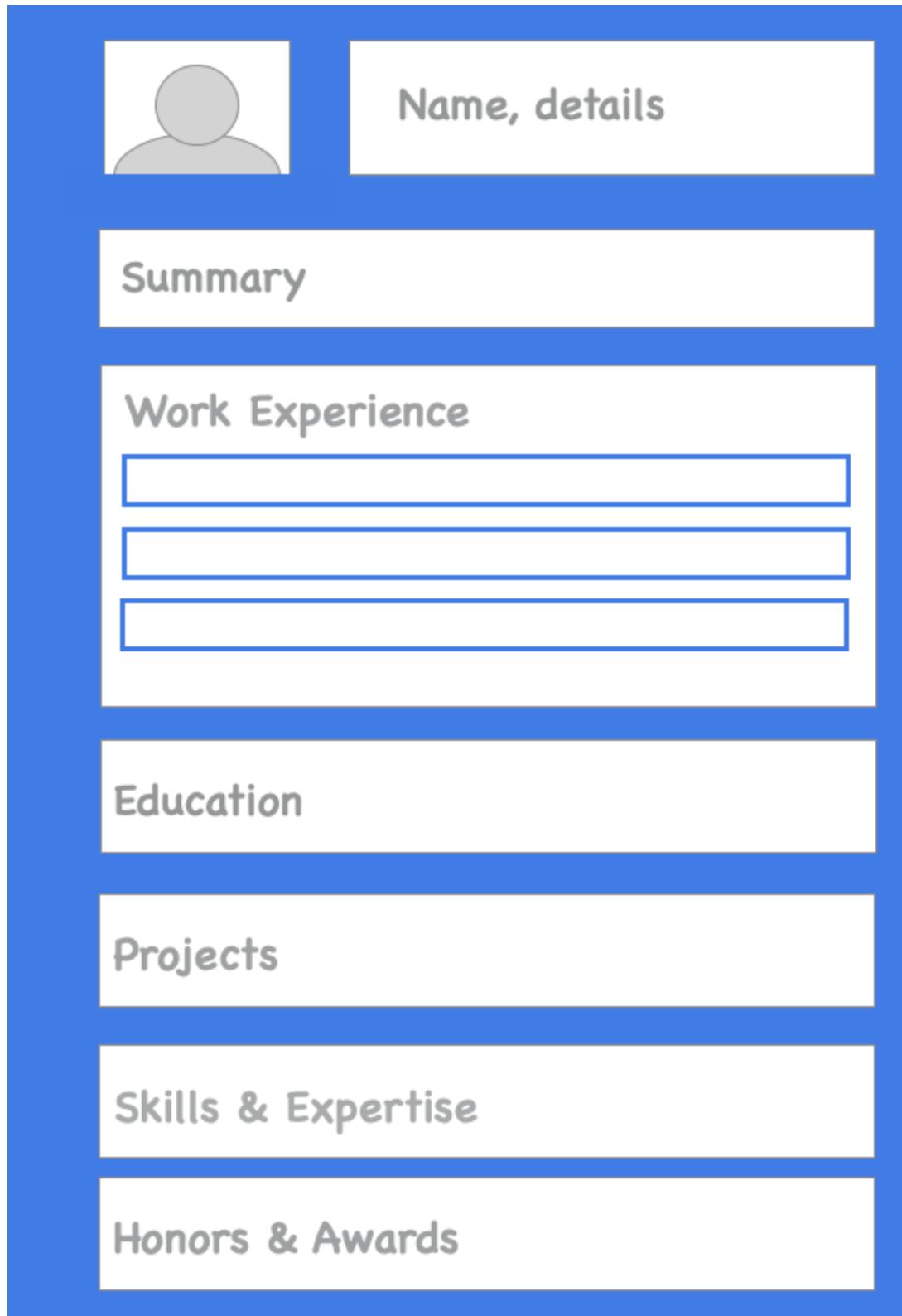
Every hiring manager, and tech recruiter I've talked with read resumes top to bottom. This includes myself. It's not just a habit: it's also a recommendation. One of the most hands-on hiring manager guide in tech is the book [Hiring Geeks that Fit](#). Here author Johanna Rothman suggests the same:

*"Start reading each resume at the top. I read résumés by starting at the top, and work my way down to the final line. (...) Seeing what a candidate thinks are his or her strong points can tell you a lot about the candidate. (...) As I read, I make a mental note that I later will compare with the actual job's requirements in each of the following areas: work experience, position objective, strengths, education, and professional training. I use items in the résumé to include candidates, and only use items to exclude candidates if the items match my elimination criteria."*

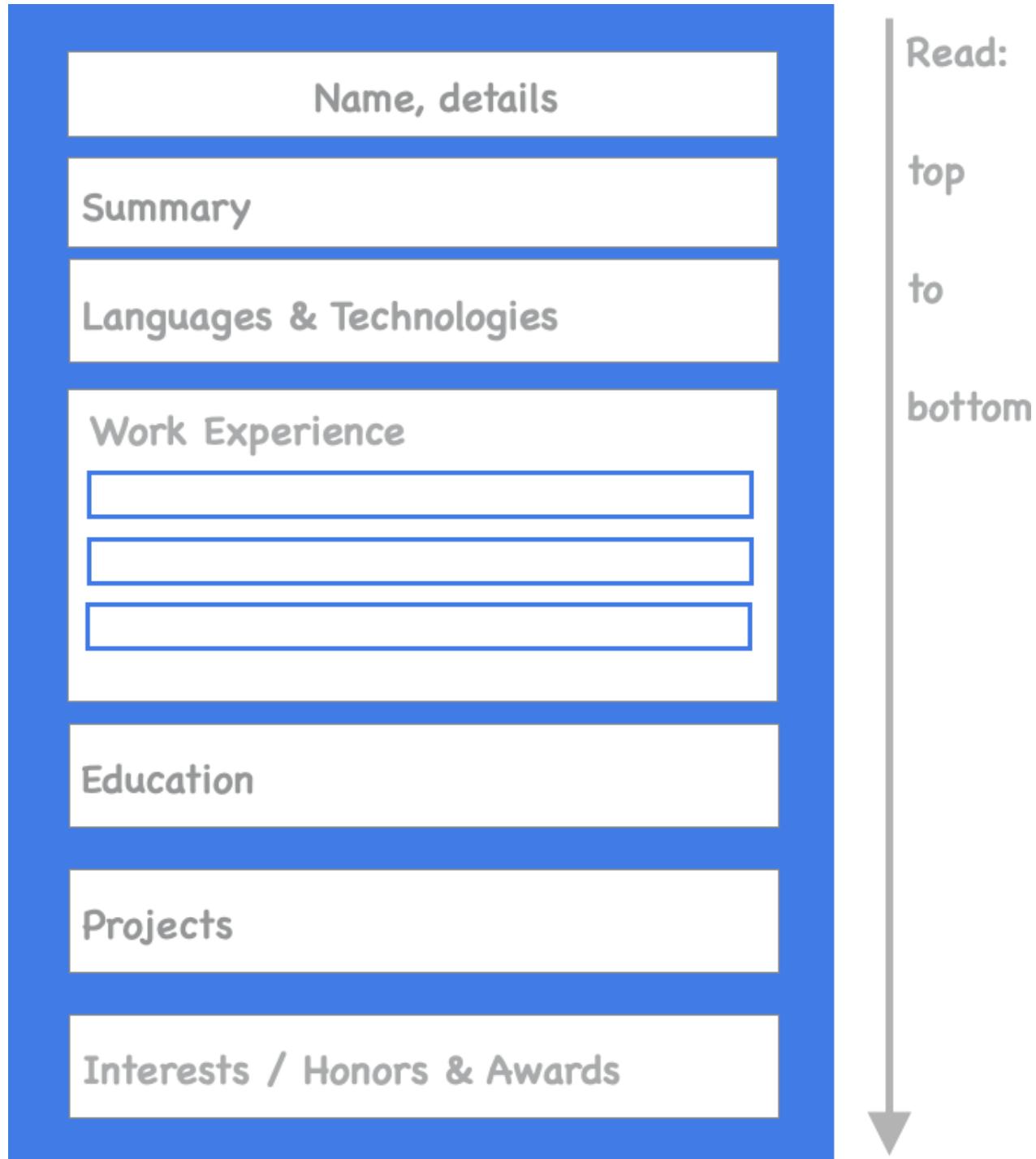
Let's look at the two most popular layouts, and how they help with these principles.

## The Top-Down Layout

The format that all recruiters in tech will be vastly familiar with is the LinkedIn profile format. They read profiles like this day in, day out. Here's what the LinkedIn profile layout looks like when recruiters view candidates:



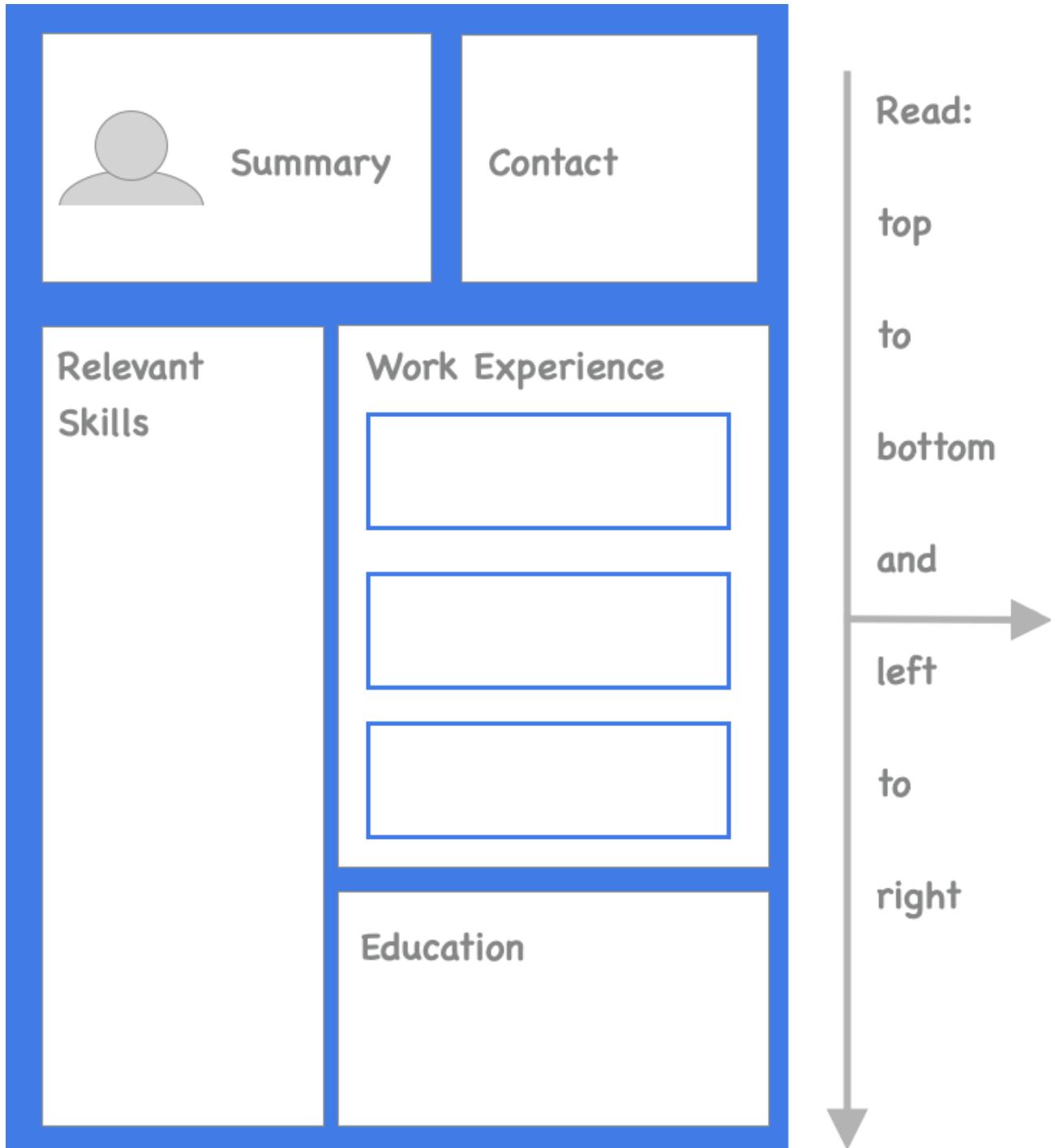
Resume templates that follow this format in their layout are a safe bet. Do keep in mind that LinkedIn follows a generic format for resumes. For software developers, languages & technologies are especially important. This is something that LinkedIn won't place emphasis on, but you should, in your resume. An adjusted format that works better for developers, yet be familiar in layout for recruiters is this one:



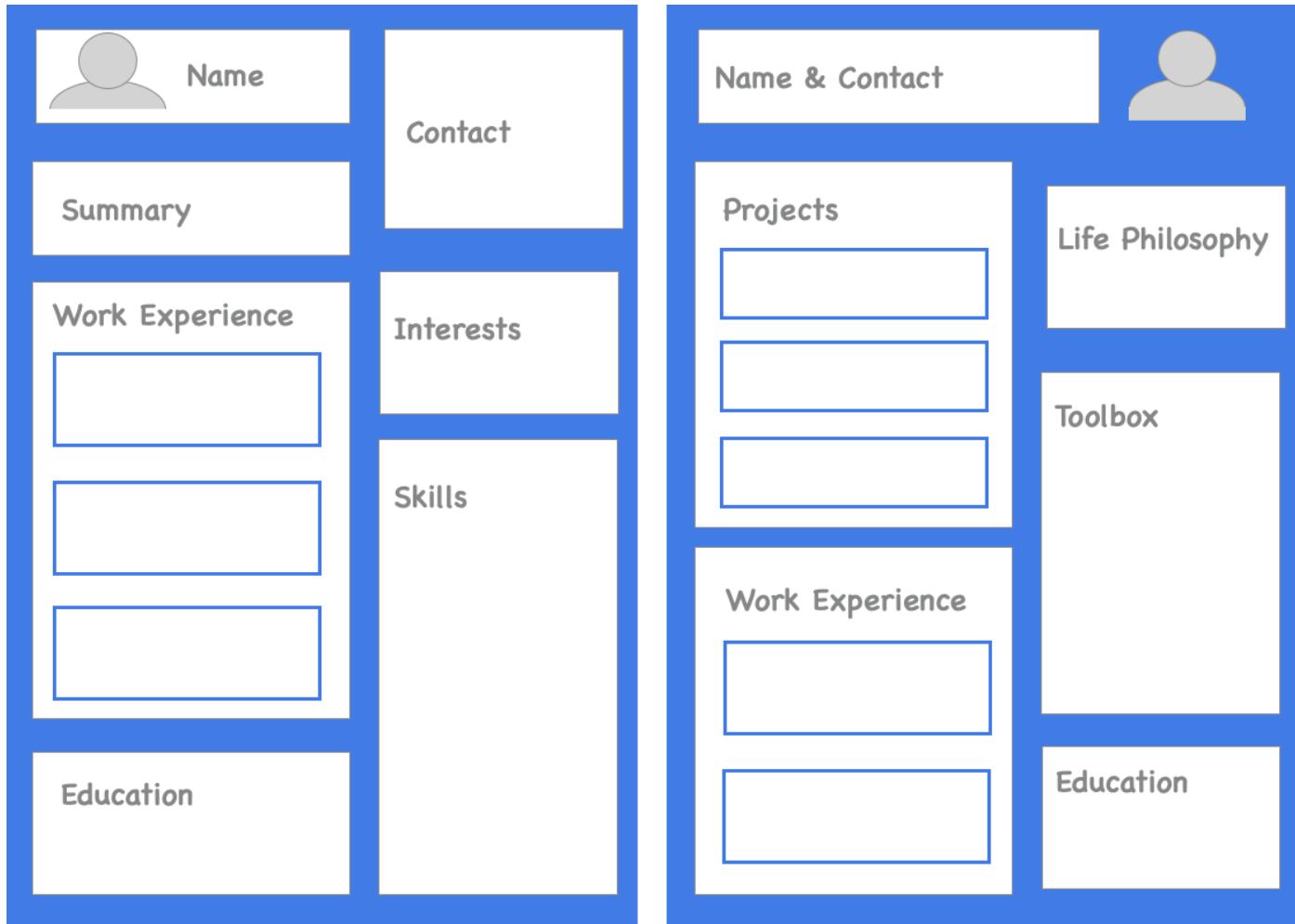
Top-down layouts work well for multi-page resumes as well. The only possible downside of this format is that it seemingly does not maximize the space on a page as efficiently as a multi-column layout. Still, given how familiar and easy to read this format is, there is little reason not to use it for your resume.

## Two-Column Layouts

Another popular layout is a two-column one. The goal of this layout is to maximize space for one-page resumes. Here's how a layout could look:



While this template seemingly does maximize space, it does this at the expense of readability. The reader needs to read top to bottom, then start again, going left to right. Also, in this format, it becomes confusing for recruiters to know where to look for key information like education or work experience. Ten resumes with two-column layouts will have their columns in ten different orders. Take a look at these two two-column resume examples, both popular layouts:



The columns are in a different order, even compared to the first example. This makes it harder for recruiters to find the key pieces of information that they are looking for—and that is so easy to spot in the top-to-down resume format.

**The biggest drawback of the two-column resume is how it discourages sharing sufficient details on your experience, both for work experience and for projects.** None of the two-column designs cater for sharing several bullet points per position. None of the two-column layouts incentivize going beyond one page for your resume. This resume format is more likely to be filled with buzzwords and cliches than to describe the impact of your work with the technologies you use.

Two-column layouts can be a great choice in cases where your resume layout is part of the application, such as for design or UX positions. In industries outside of software engineering, they might also work. However, for software developers, there is no advantage to using this layout—you'll only experience drawbacks if choosing this format.

#### From the inside out: resume layouts from a technical recruiter's point of view

Chukwuemeka Ugorji is a technical recruiter who has worked at Facebook and Andela. He has seen thousands of resumes with various formats and gives his take on the benefits and drawbacks of the top-down and two-column resumes:

*"I consider a resume a brief story of your career or maybe a 10 seconds 'elevator pitch'. If you can tell the story right in 10 seconds, then you get the time to discuss more."*

**I consider the top-down format I the most optimal** for two reasons. First, it is the most organized format for easy readability. Second, it allows for easy flow of thought. It starts with what a recruiter/hiring manager is most curious about: skills & relevant experience. Hence, this format immediately answers the key questions that the hiring team has for any candidate.

**The two-column layout is a double-edged sword.** You need to be brief with this format. If you succeed to show that you are relevant based on your skills and work experience, you will be an easy pick candidate to advance. If not, the decision to reject will be an immediate one.

*This format is like an infographic. It saves the time of the recruiter or hiring manager. However, it can hurt candidates badly if they aren't great at presenting relevant information succinctly."*

## How Recruiters and Hiring Managers Scan Resumes

Before jumping into analyzing specific resume templates, let's understand how recruiters and hiring managers scan resumes. In the first few seconds, they want to get a few key details from a resume:

- **Years of experience and dates** of your employment.
- **Key technologies** that are relevant for the role, and whether you have some proficiency with them.
- **Titles and companies** where you worked.
- **Location** where you are based right now. Are you local? Within an area that needs no visa? Somewhere where you'd likely need a visa and/or relocation?

Let's visualize what recruiters actually see from your resume in the first few seconds. Here is the resume:

## Edmond Smith

Mailing Address: TNC NT1, Cape Town  
E-Mail: edmondsmith88@example.com  
Website: <https://EdmondSmithDev.github.io/>

### TECH STACKS

- Programming Languages/Frameworks: PHP/Laravel, JavaScript/Express, Java/Kotlin for Android
- Database: Postgres, MySQL, MongoDB. Infrastructure: AWS
- Testing Tools: , PHPUnit, Mockery. API Specs: REST

### WORK EXPERIENCE

#### Cloudless, Backend Software Engineer(Remote)

Sep 2018 – present

- Re-architected and maintaining a RESTful API using PHP/Laravel with third-party API integrations such as zoom-rooms to enrich customer experience.

#### MennoMark, Team Lead

Sep 2017 – Aug 2018

- Managed a team of three developers on the development team.
- Trained and on-board two new software developers on the software products(MennoMarker, MennoMarket).
- Built a mobile app(MennoInnova) for educating farmers enrolled in the Farmer Business Management programme. This was built with Kotlin, and SQLite Database.
- Designed and implemented an API for interfacing the MennoInnova mobile application using PHP/Lumen, AWS(RDS, Elasticbeanstalk), PostMan.

#### MennoMark, Software Developer -

Jan 2018 – Mar 2018

- Built an internal mobile app(MennoMini) for recording biometric data of poultry farmers in South Africa. This was built with Java, and SQLite Database.
- Designed and implemented an API for interfacing the MennoMini mobile application using PHP/Lumen, AWS(RDS, Elasticbeanstalk), PostMan.

### VOLUNTEER WORK

#### Facebook Developer Circles Cape Town, Dev Community Lead Aug 2016 – present

- Managing an online user group for software developers.
- Organizing meetups, workshops and hackathons for software developers in Cape Town.

#### Cape Town Tech Meetups, Technical Speaker

Aug 2017

- Gave a talk on building microservices with Java Spring Boot.

### EDUCATION

#### Cape Town University – Cape Town, South Africa

Sep 2013 – Jun 2017

- Bachelor of Science in Computer Science

### AWARDS AND ACHIEVEMENTS

- Developer Community Work Spotlight - AdvancedLearning
- MasterCard Foundation Scholar – Cape Town University

Dec 2018

2013 – 2017

### SKILLS AND INTERESTS

- Experience with architecture and design principles, building to consider maintainability, performance, security requirements, and impact.
- Experience working with software engineering collaboration tools like Git, Trello, Slack.
- Working knowledge of Object-Oriented Programming concepts i.e Classes and objects, Inheritance, Encapsulation, and Polymorphism

### PERSONAL PROJECTS

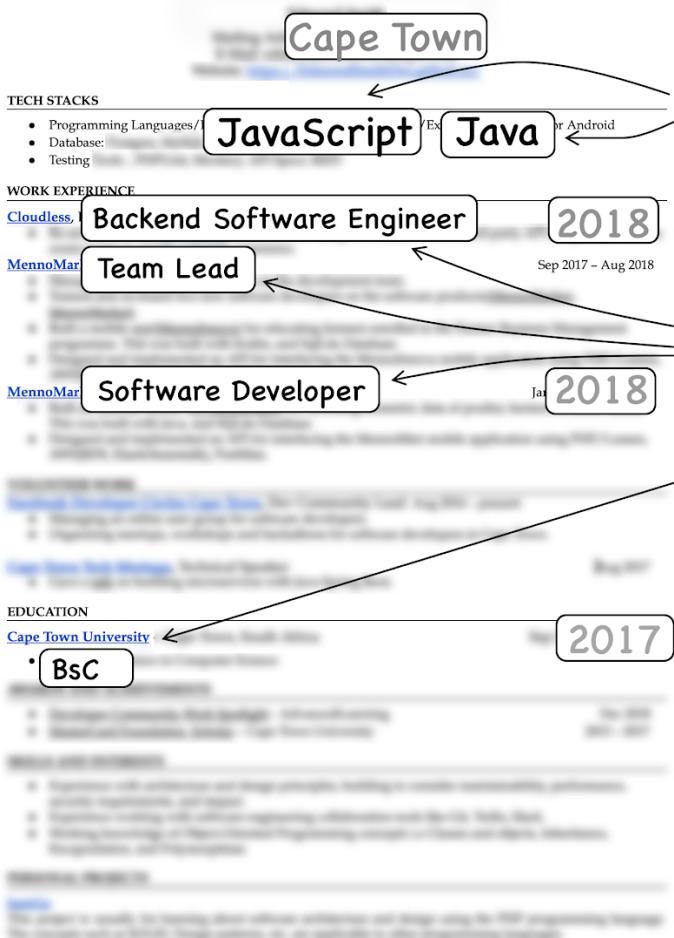
#### bareGo

This project is usually for learning about software architecture and design using the Go programming language. The concepts such as SOLID, Design patterns, etc, are applicable to other programming languages.

Here is what the recruiter will scan for in the first few seconds:



From this first scan, the recruiter notes how the candidate has at least two years' experience. Assuming this matches the guidance for experience from the hiring manager, they'll keep scanning. In the next part of the scan, they will glance to see if there are relevant technologies listed, and take a glance at the titles:



**3. Technologies**  
 Do they overlap with what the hiring manager is looking for?

**4. Titles & companies**  
 What's the career progression?  
 Any standout companies or education?

After this scan, the recruiter has the following details:

- Location of where you are based right now
- Years of experience
- Key technologies that are relevant for the role
- Titles and companies where you worked

If most of these match what the job requires, the recruiter will now start to read your resume, top to bottom. They will still only do a short scan, looking for anything relevant for the role, or that grabs their attention:



## 5. Few second scan, top to bottom

"How good of a fit would this person be for this role?"

In this case: a variety of experiences. Both mobile & backend experiences. Mentions of some best practices.

**Only after this scan would the recruiter or hiring manager look at the second page, or read more in-depth.** This assumes that what they have seen matches what they are looking for—or they've found enough standout information in other areas. These would indicate this candidate could be worth spending more time on, even if they don't have every requirement that the hiring manager is looking for.

**A good template simply makes these key pieces of information very easy to find**—and it can also guide the attention of the recruiter to a few additional pieces of information. This means that the recruiter gets all this information much quicker, and they can spend more time *actually* reading the information you want to share with them.

A good resume is a combination of an easy-to-read template and easy-to-digest content specific to the role you are applying for.

# The Results of Using a Good Resume Template

Let's take a look at how a good template can help visualize key pieces of information. Here is the updated resume, using a different top-to-bottom template. The resume contents have also been updated, applying the practices in this book, like being more specific on the impact of the candidate's work.

## Edmond Smith

Cape Town, South Africa

edmondsmith88@example.com

[Website](#)

### TECHNOLOGIES AND LANGUAGES

- PHP, JavaScript, Java, Kotlin, Android, Angular, Laravel, Express.js, Ktor
- MySQL, MongoDB, AWS, Git
- Data structures and algorithms, API design, engineering best practices, unit testing

### WORK EXPERIENCE

**Backend Software Engineer**                                      **Cloudless**    **Sep 2018 – present**

Remote

- Improved customer conversion by 30% of the [Whack co-working platform](#) by implementing recurring payments and integrating Zoom rooms.
- Re-architected the restful API powering the mobile client for [Whack](#), using PHP and Laravel.

**Development Team Lead**                                      **MennoMark**    **Jan 2017 – Aug 2018**

- Led a team of 3 engineers to ship multiple apps and services.
- Built an [app](#) for educating cocoa farmers enrolled in the Farmer Business School programme, using Kotlin and Room database.
- Designed and built a [mobile app](#) to record biometric data of poultry farmers in South Africa, using Java, and SQLite database.
- Designed and implemented APIs that powered our 10 mobile applications using PHP, Lumen, Laravel, AWS RDS, AWS Elasticbeanstalk and PostMan.
- Onboarded and trained 6 new employees on the company's development and product stack.

### EDUCATION

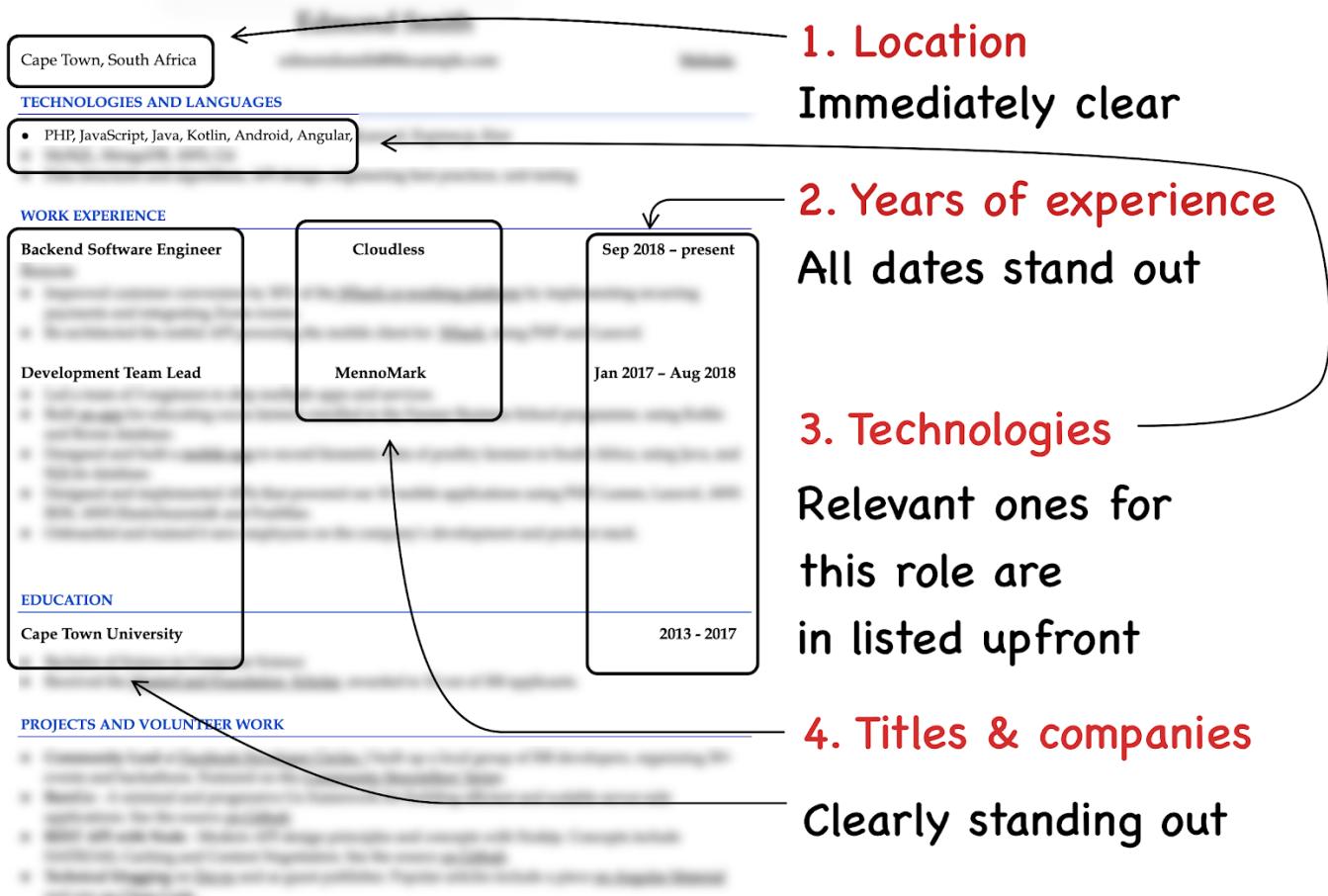
**Cape Town University**    **2013 - 2017**

- Bachelor of Science in Computer Science
- Received the [MasterCard Foundation Scholar](#), awarded to 10 out of 300 applicants.

### PROJECTS AND VOLUNTEER WORK

- **Community Lead** at [Facebook Developer Circles](#). I built up a local group of 500 developers, organizing 30+ events and hackathons. Featured on the [Community Storytellers' Series](#).
- **BareGo** - A minimal and progressive Go framework for building efficient and scalable server-side applications. See the source [on Github](#).
- **REST API with Node** - Modern API design principles and concepts with Nodejs. Concepts include HATEOAS, Caching and Content Negotiation. See the source [on Github](#).
- **Technical blogging** on [Dev.to](#) and as guest publisher. Popular articles include a piece [on Angular Material](#) and one [on Clean Code](#).

In the first scan, the recruiter will look for the same pieces of information as before. However, these are all much easier to find, as they stand out clearly:



The recruiter does not need to actively search for any of this information and they get all of this at the first glance. This means that they have more time to spend on actually reading the resume. And, as the content is well written, this is an easier task as well:

Cape Town, South Africa

**TECHNOLOGIES AND LANGUAGES**

- PHP, JavaScript, Java, Kotlin, Android, Angular, Laravel, Express.js, Ktor
- MySQL, MongoDB, AWS, Git
- Data structures and algorithms, API design, engineering best practices, unit testing

**WORK EXPERIENCE**

Backend Software Engineer      Cloudless      Sep 2018 – present

**Improved conversion by 30%**

**Re-architected**  
Development Team Lead      MennoMark      Jan 2017 – Aug 2018

**Led a team of 3**

**Built an app      Kotlin      Java      SQLite**

**Designed API used by 10 apps**

**PHP      AWS**

EDUCATION      Cape Town University      2013 – 2017

**Scholarship that 10 out of 300 got**

**PROJECTS AND VOLUNTEER WORK**

- Community Lead at E
  - BareGo - A
  - REST API with Node
  - Technical blogging or
- 500 developers, 30+ events**
- Scalable server-side apps**
- HATEOAS, Caching, Content Negotiation**
- Article on Angular & on Clean Code**

## 5. Few second scan, top to bottom

"How good of a fit would  
this person be for this role?"

In this case: impact is  
clear from the first few words.  
Numbers draw attention.  
Easy to read the resume,  
top to bottom.

**The difference between a resume with a good template and good content and one without** is how likely it is that the recruiter will get the information you are trying to convey. When you use a format that is harder to read, the recruiter will have to work hard. This means that your chances will be down to luck. If the recruiter reads your resume as one of the first of 100+ applications, they will probably take more time. However, if it comes later, they might just set it aside, as they are getting tired and are under time pressure.

With a well-written resume, even recruiters that are tired or in a hurry will get the information they need. With that, let's look at specific resume templates: both ones that we'd recommend developers to use, as well as analyzing why some popular ones do not work that well.

---

## Recap: Good Resume Template Principles

In this chapter, we went through what characteristics good developer resume templates share, and why. When choosing a resume template, aim to have them satisfy these principles:

- **Top to bottom reading.** Easy-to-read templates can be scanned in one go, from top to bottom.
- **Single column layout.** Two-column and multi-column layouts can look neat, but they are more difficult for recruiters to read. These layouts also make it harder to share enough information about your work experience and projects. A single-column layout has none of these drawbacks.
- **Important things first,** less important things towards the end. Things relevant for the position should be towards the top; choose sections accordingly.
- **Strategic use of colors and bolding.** Use bolding and colors to draw attention—and do this consistently.

# Chapter 12: Resume Templates

A template is great help in getting started with a resume. It's good practice to start with a tried and proven template, then customize it to your own style and preferences. In this section, we'll review popular templates among developers—and analyze them with pros, cons and ways to improve the template, should you choose to use it.

There is no such thing as a perfect template. And a template is not worth much without great content—content that you will need to write.

## Resume Generators vs Resume Templates

There are several resume generator sites, like resume.io, EnhanCV, VisualCV, Resume Now, Europass and many more. Apart from Europass, all of these are paid solutions. In return for your money, they promise to generate resumes that are more professional and are proven to work. Their value proposition is that by using these services, you will increase your chances of getting your position.

**Unfortunately, almost all resume generator sites work poorly for software developer resumes: and this is because they are optimized for resumes for everyone.** Most sites try to cater from finance professionals to accounting, to teachers all the way to sales people. Take the example of resume.io and the range of professions they claim to cater for:

### Select resume example category

All categories	277	Hospitality & Catering	17	Real Estate	10
Accounting & Finance	11	Human Resource	5	Retail	8
Administrative	9	Information Technology (IT)	10	Sales	9
Beauty & Wellness	7	Legal	6	Security & Protective Services	4
Business & Management	13	Maintenance & Repair	10	Social Work	11
Construction	3	Marketing	12	Sport & Fitness	18
Education	17	Medical	29	Transport & Logistics	8
Engineering	9	Other	27	Transportation	6
Government	3	Production	8		

Out of 277 resume templates, 10 are focused on tech, and 3 on software development. Still, as I reviewed all these templates, none were a choice I would strongly recommend as a technical hiring manager:

## Network Systems Analyst

5 stars

**Zach Cortez**  
Network Systems Analyst

**Profile**  
Highly skilled and dynamic Network Systems Analyst, bringing forth the ability to analyze and troubleshoot complex network infrastructures and organizations. Able to set up administrator and server accounts, and experienced in maintaining the functionality of systems. Resourceful, and continuously advancing my knowledge of network technologies.

**Employment History**

- Network Systems Analyst at Seattle Information Technology, Seattle (Sep 2010 – Present)
  - Maintained optimal network performance by performing network diagnostic tests.
  - Ensured network security by developing security access, monitoring, and reporting.
  - Worked to maintain the confidentiality of information and network data.
  - Proved troubleshooting to system failures and resolved the first being failing solutions.
  - Trained new employees on configuration software and maintained high functioning hardware.

**Network Systems Analyst at Skilled, Seattle (Sep 2009 – Sep 2010)**

- Provided outstanding support to clients.
- Provided training and documentation to keep the organization's information safe.
- Responsible for servers, routers, switches, network maintenance tasks, and made repairs when necessary.
- Collaborated with the IT Team to ensure that security software was properly installed at all company computers.

**Education**

- Bachelor of Computer Information Systems, University of Connecticut (Storrs) (Summer 2005 – May 2010)

**High School Diploma**

Glastonbury High School, Glastonbury (September 2004 – May 2005)

**References**

- Larry Hayes from Skilljar (206-554-4200)
- Amanda Saccoccia from Skilljar (206-554-4200)

## Programmer

5 stars

**TAYLOR COOK**  
Programmer

**INFO**

Address: 3023 1/2 Ave NW, Seattle, 98107, United States, Last Seen: Jan 2012, Last Connected: Feb 2012

Skills

- Knowledgeable About Programming, Software, and Programming
- Advanced Project Management
- Strong Analytical Thinking Skills
- Complex Problem Solving Skills
- Dedicated Team Player

Languages

- German
- English

**PROFILE**

Innovative Programmer and Internet Entrepreneur driving to make the most out of the latest web technologies. I have a passion for programming, learning, and improving my skills. I am currently working as a software developer and working with various data structures.

**EMPLOYMENT HISTORY**

Programmer, Johannes Initiative (Feb 2010 – Present)

- Monitored and analyzed systems to help businesses, enterprises, and governments easily work on some of humanity's greatest challenges.
- Developed and enhanced programs to increase accuracy and efficiency.
- Used my programming skills to ensure compliance with our standards.

Programmer, Kondrakina, Inc. (Jan 2009 – Jun 2010)

- Since founding Kondrakina, Inc. in 2004 I continue to work to build and maintain its infrastructure, offering product management, design, and development services.
- Work to continuously lead developments helping people to create, develop, and maintain their own products.
- Remain up-to-date with current events around the globe, and aim to serve the needs of the world with better ideas to come.

**EDUCATION**

Master of Computer Science, Boston College (Aug 2005 – Jun 2006)

## Web Developer

5 stars

**SHANE GOMEZ**  
WEB DEVELOPER

**INFO**

Address: 1025 1/2 Ave NW, Seattle, 98107, United States, Last Seen: Mar 2012, Last Connected: Apr 2012

Phone: (206) 660-0879

Email: shane@shane.com

Nationality: American

**SKILLS**

- Performance Optimization
- Problem-solving and Decision-making
- Analytical Thinking Skills
- Software Design and Development
- Coding and Debugging

**LANGUAGES**

- English
- Spanish

**PROFILE**

Experienced Web Developer adept in all aspects of advanced web development. A innovative problem solver with a passion for technology, I bring forth expertise in design, installation, testing and maintenance of web systems. Equipped with a solid understanding of web technologies, I can quickly learn and adapt to new environments. I am a team player who works well independently or within a team setting. I am highly organized, detail oriented, and have excellent communication and interpersonal skills. I am always looking for opportunities to learn and grow, and I am currently seeking a new challenge.

**EMPLOYMENT HISTORY**

Senior Web Developer, Milbank and Horwitz (Jan 2010 – Sep 2011)

- Worked as a producer and produced team member to design, code, test, and launch web sites.
- Managed front and back end development in the company's portfolio.
- Worked closely with clients to understand their needs and requirements.
- Created wireframes and prototypes for client review.
- Ensured website's compatibility across multiple browsers.
- Identified and resolved bugs and errors.
- Followed processes and procedures related to application methods and quality assurance.

Junior Web Developer, Milbank and Horwitz (Sep 2009 – Dec 2009)

- Worked as a producer and produced team member to design, code, test, and launch web sites.
- Managed front and back end development in the company's portfolio.
- Worked closely with clients to understand their needs and requirements.
- Created wireframes and prototypes for client review.
- Ensured website's compatibility across multiple browsers.
- Identified and resolved bugs and errors.
- Followed processes and procedures related to application methods and quality assurance.

**EDUCATION**

Master of Computer Science, University of California Los Angeles (Feb 2005 – Jun 2008)

Bachelor of Computer Science, Pepperdine University (Aug 2002 – Aug 2004)

**REFERENCES**

## Film and Video Editor

5 stars

**KELLY ATKINS**  
Film and Video Editor

(415) 399-5367

**Profile**

Experienced Film and Video Editor adept in meeting strict needs throughout the course of the process. Working with clients from all walks of life, I am able to quickly adapt to their needs. Highly organized, I am able to effectively manage multiple projects simultaneously while providing the highest level of service, quality, and customer satisfaction. Specializing in editing film trailers, commercials, weddings, spirituality, and individual short films.

**Employment History**

Film & Video Editor, Alphrite Media (Jan 2010 – Sep 2011)

- Worked on site with clients and various department members more than five editing projects.
- Edited and cut several news packages.
- Created and edited two short-film projects.
- Edited and cut several music videos.
- Implemented high-quality sound effects, voice, and music smoothly into video.

Film & Video Editor, Peter P's Visuals (Aug 2010 – Dec 2011)

- Assisted editing with other film editing professionals.
- Edited and cut several music videos.
- Edited and cut several short-film projects.
- Worked with other editing studios and film剪辑 rooms around the Bay Area for the best editing process.

**Education**

Bachelor of Media Studies (Aug 2009 – Sep 2009)

Lincoln High School (Aug 2004 – Jun 2005)

**Skills**

- Time Management Skills
- Editing Software
- Editorial Communication Skills

**References**

- Jeanie Rose (See Bio)
- John DeLaney (See Bio)

## Motion Graphics Artist

5 stars

**Bill Hughes, Car Mechanic**  
441 Rocke Bank, Las Vegas, NV 89101, United States, (702) 967-7800, bill@billhughes.com

**Profile**

Dedicated and experienced car mechanic with over 10 years of experience working in automotive. As a specialized car repair technician with a wealth of expertise working in repairing, diagnosing, and maintaining vehicles. I am a hard worker and reliable technician with a passion for working in the automotive industry. I am a quick learner and able to adjust to different situations and circumstances. Committed to optimal service and safety for my customers and myself. Proficient in optical alignment.

**Employment History**

Feb 2011 – Sep 2011

- Car Mechanic, Dean Automotive (Car Repair Service Diagnostic, Alignment, Brakes, Tires, Oil Changes, etc.)
- Worked with a team to provide quality service to our customers.
- Worked with a team to provide quality service to our customers.
- Completed my practice inspections for detailing.

Oct 2011 – Jan 2012

- Car Mechanic, Bob's Automotive (Car Repair Service Diagnostic, Alignment, Brakes, Tires, Oil Changes, etc.)
- Worked with a team to provide quality service to our customers.
- Worked with a team to provide quality service to our customers.
- Completed my practice inspections for detailing.

Mar 2012 – Oct 2012

- Car Mechanic, KAMCO Transmission & Total Car Care (Car Repair Service Diagnostic, Alignment, Brakes, Tires, Oil Changes, etc.)
- Worked with a team to provide quality service to our customers.
- Worked with a team to provide quality service to our customers.
- Completed my practice inspections for detailing.

**Education**

Aug 2011

- Automotive Technology Certificate, Lincoln Technical Institute (High School Diploma, Automa High School)

Sep 1996 – May 2002

- Automa High School (High School Diploma, Automa High School)

**Skills**

- Diagnose & Repair
- Experience with Maintenance & Repair
- Complete Overhaul
- Body Repair & Refinishing
- Paint Repair & Refinishing

**References**

- Jon Michael from Dean Automotive (Phone number: (702) 967-7800)
- Eric Margolis from Bob's Automotive (Phone number: (702) 967-7800)

## Software Developer

5 stars

**Jim Ryan**  
Software Developer

**Profile**

Experienced Software Developer adept at bringing forth expertise in design, architecture, and development. I have a passion for building software systems with a diverse and growing skill set. Proficient in various domains, languages, and embedded systems. Experienced with the latest cutting edge software development technologies. Able to work independently and as a team member.

**Employment History**

Software Developer at Acute Partners, New York (Sep 2011 – Present)

- Worked closely with Product Team to understand requirements and business specifications around Portfolio Management, analysis, and reporting.
- Effectively coded software changes and alterations based on specific design specifications.
- Identified and resolved problems using the latest in Cloud, Mobile, and Web Technologies.
- Developed automated buildings and solutions to audiences including senior executives and stakeholders.

Junior Software Developer at CyberCoders, New York (Jun 2011 – Sep 2011)

- Effectively addressed complex bugs.
- Improved the overall performance module under the direction of Senior Software Developers.
- Assisted with the software development team, while also serving as an effective and enthusiastic collaborator.
- Performed automated testing tasks and developed complex features.

**Education**

Aug 2009

- Master of Science in Computer Science, The Massachusetts Institute of Technology (Cambridge)

Sep 2005

- Bachelor of Computer Science, Dartmouth, Hanover (Summer 2005 – May 2006)

**References**

- Dr. Carl Czerniawski from The Massachusetts Institute of Technology (greenmantis@mit.edu)
- Dr. Jennifer Petros from The Massachusetts Institute of Technology (greenmantis@mit.edu)

## 3D Animator

5 stars

**Allie Ford, 3D Animator**  
1100 29th St, New York, NY 10010, allieford@alliedcg.com

**Profile**

Experienced 3D Animator with solid training in Graphic Design and Animation. Bringing forth the ability to create unique and creative designs for clients. Able to work independently and as a team member to meet deadlines and adhere to specific specifics. Accustomed to working well with others and committed to meeting deadlines and adhering to specific specifics.

**Employment History**

Jun 2011 – Sep 2011

- 3D Artist, Blue Sky Studios (Growth)
- Appeared the 3D Animators for developing experiences and environments for clients.
- Modelled characters and developed scenes in various environments.
- Used the latest in 3D rendering techniques to create unique and creative designs.
- Worked with a wide variety of clients and environments.

Oct 2011 – May 2012

- Animators to Graphic Design, TTS Network (New York)
- Worked closely with clients to understand requirements, and created designs for them.
- Used the latest in 3D rendering techniques to create unique and creative designs.
- Worked with a wide variety of clients and environments.

May 2012 – Mar 2013

- 3D Artist, Blue Sky Studios (New York)
- Effectively worked with clients to understand requirements, and created designs for them.
- Used the latest in 3D rendering techniques to create unique and creative designs.
- Worked with a wide variety of clients and environments.

**Education**

Aug 2007 – Sep 2011

- Bachelor of Graphic Design, Hunter College (High School Diploma, Leopold High School)

Sep 2001 – Aug 2007

- High School Diploma, Leopold High School (New York)

**Skills**

- Advanced Modeling Skills
- Knowledge of Character
- Knowledge of Modeling
- Knowledge of Rigging
- Knowledge of Shading

**References**

- Allie MacNish from ETTS Network (macnishes@etts.net)
- Lisa Sacher from Blue Sky Studios (lisasacher@blue-sky.com)
- Matthew Kirk from Blue Sky Studios (matthewkirk@blue-sky.com)

## IT manager

5 stars

**Tracy Willis**  
IT Manager

Address: 1000 Franklin Rd, Indianapolis, IN 46260, USA, (317) 633-1178, tracywillis@willisit.com

**Profile**

Experienced and professional IT Manager with over nine years of valuable experience in managing IT performance to achieve company success. Focused and relate to working well with clients and partners to ensure that all IT needs are met. Highly organized and detail oriented. Responsible for continually advancing all areas and under budget per project. Assured to maintain the highest level of IT performance and reliability. Able to work well with clients and partners, and with IT staff and applications. Bring forth a consistent attitude and the ability to establish strong and positive relationships with all company members.

**Employment History**

09/2002 – 09/2003

- IT Manager at Physio Biometrics Inc. (New York)
- Effectively managed and maintained all IT assets and clearly communicate specifications.
- Assisted and educated clients in creating and protecting their projects.
- Increased customer satisfaction by creating solutions for project enhancement.
- Ensured timely delivery by all IT staff and maintained punctuality and dependability for further productivity.

09/2003 – 09/2005

- IT Manager at Equal Inc. (New York)
- Increased revenue by 50% by implementing migration IT projects and ensuring aggressive budget constraints.
- Effectively managed and maintained all IT assets and clearly communicate requirements, which were presented at meetings.
- Increased customer satisfaction by creating solutions for project enhancement.
- Ensured timely delivery by all IT staff and maintained punctuality and dependability for further productivity.

**Education**

Aug 2008 – May 2012

- New York University (Bachelor of Computer Science)

Aug 2004 – May 2008

- Rhodes College (Bachelor of Arts)

# TAYLOR COOK

Programmer

## INFO

### ADDRESS

1600 Amherst Way, Palo Alto,  
94304, United States

### PHONE

(315) 802-8179

### EMAIL

taylor.cook@gmail.com

### NATIONALITY

American

## SKILLS

Performance Optimization

Troubleshooting and Solutions Deployment

Analytical Thinking Skills

Software Design and Development

Coding and Scripting

## LANGUAGES

English

French

## PROFILE

Innovative Programmer and Internet Entrepreneur striving to make the world a more unified and connected place. A creative thinker, adept in software development and working with various data structures.

## EMPLOYMENT HISTORY

### Programmer, Johannes Initiative

Palo Alto

Dec 2015 — Present

- Worked to enhance software systems to help educators, scientists, and policy experts already working on some of humanity's greatest challenges.
- Developed and enhanced programs to increase accuracy and lower costs.
- Developed strategies to ensure compliance with new standards.
- Handled debugging and troubleshooting with a high success rate.

### Programmer, Kindlinks, Inc.

Menlo Park, CA

Feb 2004 — Sep 2015

- Since founding Kindlinks, Inc. in 2004 I continue to work to build and improve its infrastructure, offerings, product strategy, and design.
- Work to continuously lead developments helping people to create, share, and discover in new ways.
- Remain up to date with current events around the globe, and aim to serve the people of the world with better ways to connect.

## EDUCATION

### Master of Computer Science, Boston College

Boston

Aug 2001 — Jun 2004

This resume template—supposedly created by professionals—has nothing to do with a software developer resume. It doesn't have a single programming language listed. Yet it has spoken languages on the template. The skills section doesn't even pass the buzzword bingo. And the template, with its two-column layout, is a poor choice for the resume.

**The criteria of a decent resume generator site** should be this:

- Resume templates optimized for hiring managers.** Are the templates created with the actual audience and easy readability in mind?
- Testimonials from real hiring managers.** While many resume generators and resume services claim that they'll help you score more callbacks, I would expect any paid solution to back this up with real hiring managers or recruiters in their industry.
- Real, successful resume examples.** Similar to testimonials from hiring managers, you should be wary of paying up, if a site doesn't showcase real—even if anonymized—resume examples that got people hired.

- 
- 4. **For paid services: clear pricing and value proposition.** Several resume sites have "bait pricing" where they advertise as free, then follow dark patterns to charge. Some have unclear pricing tiers. If and when you pay, the value proposition should be clear. What do you get for your money, beyond a PDF resume?
  - 5. **Not quoting ATS myths and other false facts.** You can identify resume sites that try to sell resume templates and services without industry expertise. These sites, typically quite things like "75% of resumes are never seen by humans" and other [ATS Myths](#) in order to upsell their service.

I've searched extensively among resume generator sites, and most do not satisfy these requirements for software developer resumes. In my search, I've uncovered one that does so: [Standard Resume](#). Not surprisingly, software engineers founded this site, built it for people in tech and in collaboration with technical hiring managers.

## Standard Resume: By Developers, for Developers

[Standard Resume](#) is a resume generator site that lacks all the marketing promises that every other site does. However, they have what no other site has: real resume examples of software developers and engineering managers who used their service to craft resumes that got them jobs at the likes of Dropbox, Ebay, Facebook, Amazon and many others.

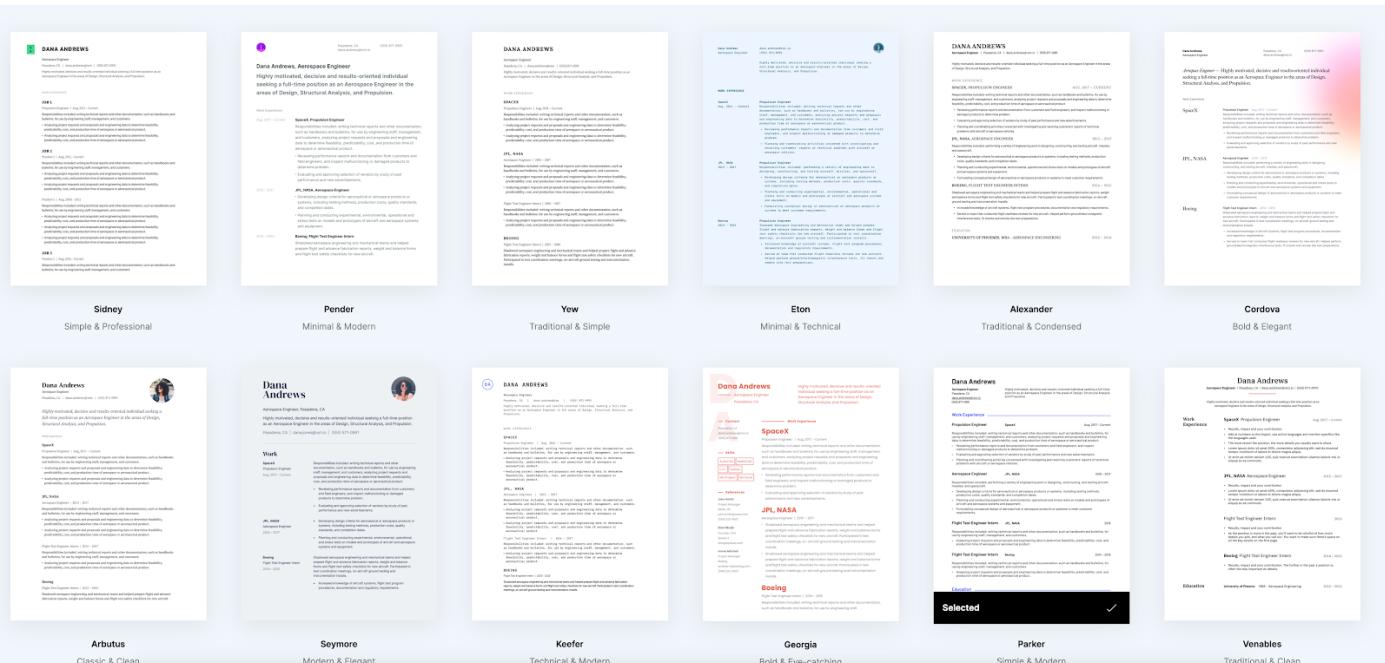
This resume site was founded by a group of techies working together at a SF startup. They used the MVP of the product to apply for jobs, and got offers from Dropbox and Metalab. In building the early version of Standard Resume, they wanted to solve for problems that no other resume generator on the market had an answer for:

- **Generate both a web resume and a pdf one.** Sending over a PDF can be a hassle, especially when you are reaching out on LinkedIn. Standard Resume was built so you can send a link that renders your resume with the same layout as the PDF.
- **Track views of your (web) resume.** How can you tell if someone has read your resume? By tracking if they visited the link you sent over.
- **Import your details from LinkedIn** to your resume to make it quicker to build your CV.

I recommend using either [Standard Resume](#), or one of the templates below to build your resume. While Standard Resume is a paid subscription, the biggest benefits of it are these:

1. **The cleanest tech resume layouts I've seen.** You cannot "cram" the fonts, as UX professionals designed spacing for the resume templates. Almost everyone plays around with font sizes on templates, and the result is often a resume that feels crammed and unprofessional. I've had several people ask me if they can make the font size smaller on one of the other templates. With this solution, you need to worry about the content, not the design.
2. **A web-only resume version** that you can send over even before formally applying for a job. If you DM people, reach out on LinkedIn, you often can't send a PDF resume. But you can send a web link that has the same contents.
3. **Web resume view tracking**, so you get a sense if someone has actually viewed the resume you shared over a link.
4. **Real resume examples of real developers and engineering managers** who actually got jobs at tech companies with that resume. Almost all other resume sites fail to show real developer resume examples of people working at well-known tech companies. With Standard Resume, you can [browse real resumes](#) for software engineers, iOS, frontend developers, data scientists, engineering managers and other professions who have successfully used the service.

It should be of little surprise that almost all Standard Resume templates are single-column and all templates have a heavy focus on spacing and typography—another thing that sets it apart from other sites. As a hiring manager, all of these layouts are easy—even pleasant—to read.



Resume templates from Standard Resume. Note that all but one follow the top-down template, with an easy-to-read structure and layout.

## From the inside out: lessons learned from the founders of Standard Resume

Riley Tomasek is a software engineer and a cofounder of Standard Resume. After users have created more than 100,000 resumes - much of this for software developers - he shares lessons he's learned about what makes for a good resume in tech:

**"One of the most common mistakes is cramming a ton of content into a single page. This is the number one issue I see with software engineering resumes. Removing the whitespace between paragraphs and sections makes your resume harder to scan and read. So do tiny font sizes."**

**The most important part of a resume is that it's easy to read.** If the recruiter skips your resume because it's hard to read, nothing else—including the contents of your resume—matters. Either focus on the critical parts or make it two pages.

**Resume typography is particularly important** when applying for jobs in tech, especially for design-heavy companies. Since resumes are mostly text, typography is responsible for a large part of your resume's first impression. Take the time to get the correct line length, font sizes, line heights, etc. — or let professionals handle it."

I came across Standard Resume while researching resume generator sites. I was about to close with the recommendation of *not* recommending any resume generators for tech resumes. However, the clean approach Standard Resume takes, and the easy to read templates made me change my mind. I reached out to the team, offering feedback on the resume templates, and listed things I missed from the existing ones. We collaborated and helped them create the Parker and Venable themes based on my existing resume templates. These two themes are ones that I would recommend for software engineers - and we'll review them in this chapter.

## Template Reviews

In the second half of the chapter, we'll review both resume templates created for the book, as well as popular resume templates. The reviews all follow this format:

- **Link** to the template or site that provides the template.
- **Rating for software developers:** assessment on how well the template follows principles for good developer templates.
- **The good:** positive highlights on the resume format. Even if you don't use the template, you could take inspiration from the good parts.
- **The bad:** characteristics that could be a disadvantage when working with the template.
- **Ways to improve it:** ideas and suggestions on either improving or customizing the template.

Keep in mind that as most of these templates are popular, people likely had success with all of them, with the right content when applying for a position where they were a match for. The *content* of your resume is more important than the template itself.

## Recommended Resume Templates

The following resume templates are all solid—and proven—choices for software engineers and engineering managers:

- The Pragmatic Engineer's resume template
- Standard Resume: Parker template
- The Mono Engineers' resume template
- Markdown to PDF resume template
- The Experienced Engineer's resume template
- Standard Resume: Venable's template
- CareerCup resume template

Let's take a closer look at each of them.

# The Pragmatic Engineer's Resume Template

[The Pragmatic Engineer's Resume Template](#) is a clean template that is a good choice to use for developers.

## Rating for software developers: ★★★★☆

### The good:

- The template has many similarities to the Career Cup template, meaning US tech recruiters will find it familiar.
- Good use of bolding and consistent formatting makes the resume easy to scan.
- Dates and titles are *really* easy to read.
- The format works both for one- and two-page resumes.
- The template works well both with and without the summary section.

### The bad:

- The template does not cater to people who would want to add more personal information. This can be customized, though.

### Ways to improve it:

- Customize the highlight color and the fonts to your preferences.

City, Country  
resume@pragmaticengineer.com

Gergely Orosz

[linkedin.com/in/gergelyorosz](https://linkedin.com/in/gergelyorosz)  
[github.com/gergelyorosz](https://github.com/gergelyorosz)

#### Work Experience

Senior Software Engineer Payments team	Uber Amsterdam, Netherlands	Sep 2018–Present
Software Engineer II Customer Success team	Skyscanner London, UK	2017–2018
Software Developer	Skype London, UK	2015–2017
Software Developer Intern	LogMeIn Budapest, Hungary	Summer 2014

• Results, impact and your contribution.  
• Add at numbers to the impact, use active languages and mention specifics like the languages used.  
• The more recent the position, the more details you usually want to share.  
• Lorem ipsum dolor sit amet 40%, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
• Ut enim ad minim veniam 100, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.

• Results, impact and your contribution  
• Lorem ipsum dolor sit amet 40%, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
• Ut enim ad minim veniam 100, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.

• Results, impact and your contribution  
• As the position is more in the past, you'll want to be mindful on how much details you add, and what you call out. You want to make sure there's space on all the key details on the first page.

• Results, impact and your contribution. The further in the past a position is, often the less important its details.

#### Education and Certifications

M.Sc. Computer Science, University of Munich, Germany. B.Sc. Computer Science, University of Technology, Budapest.	2013–2014 2010–2013
---	------------------------

#### Technologies and Languages

- Languages: Java, Go, Swift, Kotlin, {other languages}
- Technologies: MySQL, Postgres, AWS, Git, {other technologies}
- Other: Data structures and algorithms, {other relevant skills}

## Standard Resume Parker Template (Paid)

The Parker resume template from Standard Resume is the “porting” of The Pragmatic Engineer’s Resume template to this resume builder. I collaborated with the Standard Resume team in building this template. I find this template slightly more readable and better formatted than The Pragmatic Engineer’s Resume Template, thanks to the touch up from the Standard Resume design team.

### Rating for software developers: ★★★★☆

#### The good:

- The template has many similarities to the Career Cup template, meaning US tech recruiters will find it familiar to scan.
- Crafted by designers, with a heavy emphasis on readability and spacing. The clarity shows: I find this template easier to scan than The Pragmatic Engineer’s Resume Template.
- A web version with the same layout can be shared with a link.
- Web resume views are tracked by the platform.
- The template works well when applying for US-companies.

#### The bad:

- Unlike a template, it’s harder to “cram” everything on one page. But this is by design.
- Locations cannot be recorded separately for each position—I added this information to the company name field.

#### Ways to improve it:

- Customize the highlight colors with the Standard Resume color picker.

### Gergely Orosz

#### Work Experience

Senior Software Engineer      Uber (Amsterdam, NL)      Sep, 2018 - Current

- Results, impact and your contribution.
- Add at numbers to the impact, use active languages and mention specifics like the languages used.
- The more recent the position, the more details you usually want to share.
- Lorem ipsum dolor sit amet 40%, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- Ut enim ad minim veniam 100, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.

Software Engineer II      Skyscanner (London, UK)      2017 - Feb, 2018

- Results, impact and your contribution
- Lorem ipsum dolor sit amet 40%, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- Ut enim ad minim veniam 100, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.

Software Developer      Skype (London, UK)      2015 - 2017

- Results, impact and your contribution
- As the position is more in the past, you’ll want to be mindful on how much details you add, and what you call out. You want to make sure there’s space on all the key details on the first page.

Software Developer Intern      LogMeIn (Budapest, HU)      2014 - 2014

- Results, impact and your contribution. The further in the past a position is, often the less important its details.

#### Education

M.Sc. Computer Science      University of Munich (Germany)      2013 - 2014

B.Sc. Computer Science      University of Technology (Budapest, Hungary)      2010 - 2013

#### Technologies and Languages

Languages      Java, Go, Swift, Kotlin, {other languages}

Technologies      MySQL, Postgres, AWS, Git, {other technologies}

Other      Data structures and algorithms, {other relevant skills}

# The Mono Engineer's Resume Template

[The Mono Engineer's Resume Template](#) is a slight modification to the Pragmatic Engineer's Resume template.

## Rating for software developers: ★★★★☆

### The good:

- It draws more attention to either the company or the job title.
- It can be a good fit for people who either have "standout" companies and education on their resume, or for people who have companies that recruiters and hiring managers are unlikely to recognize, and thus focusing on the titles makes more sense.
- The other good parts are the same as with The Pragmatic Engineer's Resume Template.

### The bad:

- Same as with The Pragmatic Engineer's Resume Template.

### Ways to improve it:

- Add a highlight color and/or adjust fonts to your preferences.

## Gergely Orosz

resume@pragmaticengineer.com • Amsterdam, Netherlands • [LinkedIn](#) • [Github](#)

I am a full-stack software engineer especially interested in building magical mobile experiences, scaling systems up, and shipping reliable applications. Note: use the summary when it makes sense.

### WORK EXPERIENCE

**Uber** Sep 2018–Present  
Amsterdam, Netherlands

Senior Software Engineer

- Results, impact and your contribution.
- Add at numbers to the impact, use active languages and mention specifics like the languages used.
- The more recent the position, the more details you usually want to share.
- Lorem ipsum dolor sit amet 40%, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- Ut enim ad minim veniam 100, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.

**Skyscanner** 2017–2018  
London, UK

Software Engineer II

- Results, impact and your contribution
- Lorem ipsum dolor sit amet 40%, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- Ut enim ad minim veniam 100, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.

**Skype** 2015–2017  
London, UK

Software Developer

- Results, impact and your contribution
- As the position is more in the past, you'll want to be mindful of how much details you add, and what you call out. You want to make sure there's space on all the key details on the first page.

**LogMeIn** Summer 2014  
London, UK

Software Developer Intern

- Results, impact and your contribution. The further in the past a position is, often the less important its details.

### EDUCATION

**University of Munich** 2013–2014  
Munich, Germany

MsC, Computer Science

- Add details only if they are (still) relevant. Such as awards, leadership, results.

**Budapest University of Technology & Economics** 2010–2013  
Budapest, Hungary

BSc, Computer Science

### TECHNOLOGIES AND LANGUAGES

- Languages: Java, Go, Swift, Kotlin, {other languages}
- Technologies: MySQL, Postgres, AWS, Git, {other technologies}
- Other: Data structures and algorithms, {other relevant skills}

## Markdown to PDF Resume Template

[Markdown to PDF resume template generator](#) by Mike Lee Williams is a neat way to use Markdown as the source of truth for your resume and to store different versions on GitHub. The default style is similar to the Mono Engineer's Resume template, and you can customize it further.

### Rating for software developers: ★★★★☆

#### The good:

- A clean template.
- Markdown as the data source for the resume.
- Store different versions of your resume in GitHub (or another source)
- You get to build your resume from the command line!

#### The bad:

- Same as with The Mono Engineer's Resume Template.

#### Ways to improve it:

- Add a highlight color and/or adjust fonts to your preferences using the css for this template.

### RICHARD HENDRICKS

[richard.hendricks@mail.com](mailto:richard.hendricks@mail.com) • (912) 555-4321 • [richardhendricks@example.com](mailto:richardhendricks@example.com) • San Francisco, CA

CEO and Software Engineer with knowledge of applied information theory, including optimizing lossless compression schema of both the length-limited and adaptive variants.

#### EXPERIENCE

##### CEO/President, Pied Piper

Dec 2013 – Dec 2014

Pied Piper is a multi-platform technology based on a proprietary universal compression algorithm that has consistently fielded high Weisman Scores™ that are not merely competitive, but approach the theoretical limit of lossless compression.

- Build an algorithm for artist to detect if their music was violating copyright infringement laws
- Successfully won Techmunch Disrupt
- Optimized an algorithm that holds the current world record for Weisman Scores

##### Teacher, CoderDojo

July 2013 – Dec 2013

Global movement of free coding clubs for young people.

- Awarded 'Teacher of the Month'

#### PROJECTS

##### Miss Direction

Aug 2016

A mapping engine that misguides you:

- Won award at AIHacks 2016
- Built by all women team of newbie programmers
- Using modern technologies such as GoogleMaps, Chrome Extension and Javascript

#### EDUCATION

##### University of Oklahoma, BA Information Technology

2011 – 2014

- GPA 4.0
- DB1101 - Basic SQL
- CS2011 - Java Introduction

#### SKILLS

- Web development: HTML, CSS, JavaScript
- Compression: Mpeg, MP4, GIF

# The Experienced Engineer's Resume Template

[The Experienced Engineer's Resume Template](#) is a template designed as a two-page template for experienced developers with standout experience. It works well for above senior candidates (principal, staff, distinguished engineers), people with standout companies on their resume, or for people with additional standout contributions (e.g. open source projects or contributions).

## Rating for software developers:

- With standout experience: ★★★★★
- Without this experience: ★★★☆☆

## The good:

- Company name and title have stronger than usual attention.
- Dates are easy to scan.
- Links draw a lot of attention—and they do this deliberately. They are inviting the hiring manager or recruiter to click through and see strong content behind the links.

## The bad:

- Leaves unused space on the left side—however, this is deliberate, to give more focus to the company names and titles.
- Technologies & languages will most likely be on the second page. This is deliberate: it is meant for standout profiles where recruiters will scan the second page as well, being impressed with either company names, title, or both.
- This template doesn't work well for people with less experience or people with few standout areas.

## Ways to improve it:

- Make the color scheme your own.

## Gergely Orosz

[blog.pragmaticengineer.com](#)

Amsterdam, Netherlands • [resume@pragmaticengineer.com](#) • +1 23456 789

I am a full-stack software engineer especially interested in building magical mobile experiences, scaling systems up, and shipping reliable applications. Note: use the summary when it makes sense. See [The Tech Resume Inside-Out](#) for pointers.

WORK EXPERIENCE	Uber - Senior Software Engineer Amsterdam, Netherlands	Sep '18 - Present
	<ul style="list-style-type: none"><li>Results, impact and your contribution.</li><li>Add at numbers to the impact, use active languages and mention specifics like the languages used.</li><li>The more recent the position, the more details you usually want to share.</li><li>Lorum ipsum dolor sit amet 40%, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</li><li>Ut enim ad minim veniam 100, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.</li></ul>	
	Skyscanner - Software Engineer II London, UK	2017 - 2018
	<ul style="list-style-type: none"><li>Results, impact and your contribution</li><li>Lorum ipsum dolor sit amet 40%, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</li><li>Ut enim ad minim veniam 100, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.</li></ul>	
EDUCATION	Msc Computer Science University of Munich. GPA: 4.0.	2013 - 2014
	Bsc Computer Science Budapest University of Technology & Economics. Top 2% of the class.	2010 - 2013
OPEN SOURCE & WRITING	<p>Open source contributions to high-profile projects:</p> <ul style="list-style-type: none"><li>Swift core contributions - see the <a href="#">20+ contributions on GitHub</a></li><li>React core contributions - see them <a href="#">on GitHub</a></li><li>Multiple RIBs contributions - see <a href="#">on GitHub</a></li></ul> <p>Technical books:</p> <ul style="list-style-type: none"><li>Co-author of <a href="#">Swift In Depth</a></li><li>Technical reviewer on <a href="#">Distributed Systems Demystified</a> and <a href="#">React from the Inside-Out</a></li></ul> <p>Writing: I publish technical articles on the Pragmatic Engineer blog. Popular ones include one on <a href="#">distributed architecture concepts</a> and another one on <a href="#">software architecture</a>.</p>	
TECHNICAL SKILLS	<p>Languages: Java, Go, Swift, Kotlin, {other languages}</p> <p>Technologies: MySQL, Postgres, AWS, Git, {other technologies}</p> <p>Other: Data structures and algorithms, {other relevant skills}</p>	

## Standard Resume Venables Template (Paid)

The [Venables resume template](#) from [Standard Resume](#) is the “porting” of The Experienced Engineer’s Resume template to this resume builder. As with the Parker template, I also collaborated with Standard Resume in building it. I’m pleased with the result, which has better spacing than The Experienced Engineer’s Resume Template and more color customization options.

### Rating for software developers:

- With standout experience: ★★★★★
- Without this experience: ★★★☆☆

### The good:

- Same as with the Experienced Engineer’s template.
- You need to fiddle less with tables than when using the Experienced Engineer’s template, and you can’t mess up the layout of this template.
- A web version with the same layout can be shared with a link.
- Web resume views are tracked by the platform.

### The bad:

- Locations cannot be recorded separately for each position—I added this information to the company name field.

### Ways to improve it:

- Customize the highlight colors with the Standard Resume color picker.

### Gergely Orosz

Work Experience	Uber (Amsterdam, NL) Senior Software Engineer	Sep, 2018 - Current
	<ul style="list-style-type: none"><li>Results, impact and your contribution.</li><li>Add at numbers to the impact, use active languages and mention specifics like the languages used.</li><li>The more recent the position, the more details you usually want to share.</li><li>• Lorem ipsum dolor sit amet 40%, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</li><li>• Ut enim ad minim veniam 100, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.</li></ul>	
	Skyscanner (London, UK) Software Engineer II    2017 - Feb, 2018	
	<ul style="list-style-type: none"><li>Results, impact and your contribution</li><li>• Lorem ipsum dolor sit amet 40%, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</li><li>• Ut enim ad minim veniam 100, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo.</li></ul>	
	Skype (London, UK) Software Developer    2015 - 2017	
	<ul style="list-style-type: none"><li>Results, impact and your contribution</li><li>As the position is more in the past, you’ll want to be mindful on how much details you add, and what you call out. You want to make sure there’s space on all the key details on the first page.</li></ul>	
	LogMeIn (Budapest, HU) Software Developer Intern    2014 - 2014	
	<ul style="list-style-type: none"><li>Results, impact and your contribution. The further in the past a position is, often the less important its details.</li></ul>	
Education	University of Munich (Germany) M.Sc. Computer Science	2013 - 2014
	University of Technology (Budapest, Hungary) B.Sc. Computer Science	2010 - 2013
Technologies and Languages	Languages	Java, Go, Swift, Kotlin, {other languages}
	Technologies	MySQL, Postgres, AWS, Git, {other technologies}
	Other	Data structures and algorithms, {other relevant skills}

# CareerCup Resume Template

The [CareerCup resume template](#) is common to see used both for Silicon Valley software engineers and people applying for tech companies in the US. The template was created by Gayle Laakmann McDowell, author of [Cracking the Coding Interview](#).

## Rating for software developers: ★★★★☆

### The good:

- Tech recruiters in the US are used to it.
- Consistent formatting of dates and company/school names across sections.
- Dates and titles are *really* easy to read. Figuring out how much experience you have is a breeze.
- The format can be used for two pages as well—just make sure the most important details that recruiters need to see are listed on the first page.

### The bad:

- Dated style, using no colors. In 2014, when this template was created, resume printing was still a thing. This template has too much black-and-white.
- Geared towards top tech companies in the US. For other companies and outside the US, you'll need to tweak this.
- The format incentivizes crowding content on the same page.

### Ways to improve it:

- If you are applying for non-US tech companies, consider moving up the Languages and Technologies section to be on the first page.
- The resume is dated in its black-and-white style. You could use colors to have headings stand out and reduce the bolding.

123 Spruce St, Apt 35 Philadelphia PA 19103	<b>GAYLE L. McDOWELL</b>	(555) 555-1212 gayle@careercup.com
<hr/>		
<b>EMPLOYMENT</b>		
<b>Software Engineer, Intern</b> iChat AV	<b>Apple Computer</b>	<b>Summer 2004</b>
• Reduced time to render the user's buddy list by 75% by implementing prediction algorithm.		
• Implemented iChat integration with OS X Spotlight Search by creating tool which extracts metadata from saved chat transcripts and provides metadata to a system-wide search database.		
• Redesigned chat file format and implemented backwards compatibility for search.		
<b>Lead Student Ambassador</b>	<b>Microsoft Corporation</b>	<b>Fall 2003 – Spring 2005</b>
• Promoted to Lead Student Ambassador in Fall 2004; supervised 10 – 15 Student Ambassadors.		
• Created and taught Computer Science courses: CSE 099: Software Design and Development.		
<b>Head Teaching Assistant</b>	<b>University of Pennsylvania</b>	<b>Fall 2001 – Spring 2005</b>
• Courses: Advanced Java III, Software Engineering, Mathematical Foundations of Computer Science I & II.		
• Promoted to Head TA in Fall 2004; led weekly meetings and supervised four other TAs.		
<b>Software Design Engineer, Intern</b> Visual Studio Core (Summer 2003)	<b>Microsoft Corporation</b>	<b>Summers 2001 – 2003</b>
• Implemented a user interface for the VS open file switcher (ctrl-tab) and extended it to tool windows.		
• Created service to provide gradient across VS and VS add-ins. Optimized service via caching.		
• Programmer Productivity Research Center (Summers 2001, 2002)		
• Built app to compute similarity of all methods in a code base, reduced time from $O(n^2)$ to $O(n \log n)$ .		
• Created test case generation tool which creates random XML docs from XML Schema.		
<b>EDUCATION</b>		
<b>Philadelphia, PA</b>	<b>University of Pennsylvania</b>	<b>Fall 2000 – May 2005</b>
• M.S.E. in Computer and Information Science, May 2005. GPA: 3.6		
• B.S.E. in Computer Science Engineering with Minor in Mathematics, May 2005. In-major GPA: 3.4.		
• Graduate Coursework: Software Foundations; Computer Architecture; Algorithms; Artificial Intelligence; Comparison of Learning Algorithms; Computational Theory.		
• Undergraduate Coursework: Operating Systems; Databases; Algorithms; Programming Languages; Comp. Architecture; Engineering Entrepreneurship; Calculus III.		
<hr/>		
<b>TECHNICAL EXPERIENCE</b>		
<b>Projects</b>		
• <b>Multi-User Drawing Tool</b> (2004). Electronic classroom where multiple users can view and simultaneously draw on a “chalkboard” with each person’s edits synchronized. C++, MFC		
• <b>Synchronized Calendar</b> (2003 – 2004). Desktop calendar with globally shared and synchronized calendars, allowing users to schedule meetings with other users. C# .NET, SQL, XML		
• <b>Operating System</b> (2002). UNIX-style OS with scheduler, file system, text editor and calculator. C		
<hr/>		
<b>ADDITIONAL EXPERIENCE AND AWARDS</b>		
• <b>Instructor</b> (2003 – 2005): Taught two full-credit Computer Science courses; average ratings of 4.8 out of 5.0.		
• <b>Third Prize, Senior Design Projects:</b> Awarded 3rd prize for Synchronized Calendar project, out of 100 projects.		
<hr/>		
<b>LANGUAGES AND TECHNOLOGIES</b>		
• C++, C, Java, Objective-C; C# .NET, SQL, JavaScript, XSLT, XML (XSD) Schema		
• Visual Studio; Microsoft SQL Server; Eclipse; XCode; Interface Builder		

## Other Resume Templates

There are other resume templates that are not great choices for a developer or engineering manager resume for different reasons. We'll look at the ones that I've seen most frequently used by people:

- Google Docs: Serif resume template
- Google Docs: Swiss resume template
- Canva resume templates
- Resume.io: software engineer template
- Resume.io: web developer template with skill points
- VisualCV: Standard template
- EnhanCV: software engineer template
- Europass CV

Although I hesitate to recommend these templates, they could work well enough when addressing their drawbacks, or for certain use cases. Don't forget that the resume contents matter more than the template itself.

Let's review the templates, analyzing both the issues, as well as their strengths.

## Google Docs Serif Resume Template

I am not a fan of two-column resumes, as they make it hard to scan from top to bottom and they squeeze the space you can add for your experiences. However, if you go with one, the [Google Docs Serif resume](#) is still an okay choice. It is not specific to software development, so I would change "Skills" to "Technologies" and remove the Awards / Languages.

### Rating for software developers: ★★★☆☆

#### The good:

- Clean format and nice use of colors.
- A one-page resume format.

#### The bad:

- It's harder for recruiters to scan and answer the question of "how much experience do you have?" Dates are hard to find.
- You don't have much space to write about your experience and the impact you made.
- For a two-page resume with more experience, this format doesn't work as well.
- It's not specific for tech. "Awards" and "Languages" make little sense.

#### Ways to improve it:

- Only use this template if you can definitely fit everything on one page.
- Customize the right column and make good use of it. "Awards" are rarely relevant for tech resumes. Look at tips on improving the EnhanCV software engineer resume.

## Your Name

123 Your Street  
Your City, ST 12345  
(123) 456-7890  
no\_reply@example.com

### EXPERIENCE

#### Company, Location — Job Title

MONTH 20XX - PRESENT

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonumy nibh.

#### Company, Location — Job Title

MONTH 20XX - MONTH 20XX

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonumy nibh.

### AWARDS

#### Company, Location — Job Title

MONTH 20XX - MONTH 20XX

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonumy nibh.

### EDUCATION

#### School Name, Location — Degree

MONTH 20XX - MONTH 20XX

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonumy nibh euismod tincidunt ut labore dolore.

#### School Name, Location — Degree

MONTH 20XX - MONTH 20XX

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonumy nibh.

### PROJECTS

#### Project Name — Detail

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

123 Your Street  
Your City, ST 12345  
(123) 456-7890  
no\_reply@example.com

### SKILLS

Lorem ipsum dolor s

Consectetuer adipisc

Sed diam nonummy

euismod tincidunt.

Laoreet dolore magn

aliquam erat volutpa

### AWARDS

Lorem ipsum dolor :

Consectetuer adipisc

Sed diam nonummy

Nibh euismod tincid

laoreet dolore magn

erat volutpat.

### EDUCATION

Lorem ipsum dolor :

Consectetuer adipisc

Sed diam nonummy

Nibh euismod tincid

laoreet dolore magn

erat volutpat.

### LANGUAGES

Lorem ipsum, Dolor

Consectetuer

# Google Docs Swiss Resume Template

While I do not recommend using the [Google Docs Swiss resume](#) as is. What I like about this template is the order of the sections. Otherwise, this template is not a good choice for a software developer's resume.

## Rating for software developers: ★★★☆☆

### The good:

- Order of sections: skills (technologies/languages), work experience, then education. This order works well for software engineers.

### The bad:

- Lots of space wasted on the left side.
- Hard for recruiters to scan the dates for your work experience and education.

### Ways to improve it:

- I advise not to use this template for software engineering positions.
- For experienced engineers, consider using the Experienced Engineer's Template or Standard Resume Venables template. Both offer a similar layout but better formatting.

## Your Name Creative Director

Your Name  
123 Your Street  
Your City, ST 12345  
123.456.7890  
no\_reply@example.com

### Skills

—  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean interdum nisi. Sed in consequat mi. Sed pulvinar lacinia felis eu f

### Experience

—  
Company Name / Job Title  
MONTH 20XX - PRESENT, LOCATION  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean interdum nisi. Sed in consequat mi. Sed in consequat mi, sed pulvinar lacinia felis eu finibus.

### Education

—  
Company Name / Job Title  
MONTH 20XX - MONTH 20XX, LOCATION  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean interdum nisi. Sed in consequat mi. Sed pulvinar lacinia felis eu f  
  
School Name / Degree  
MONTH 20XX - MONTH 20XX, LOCATION  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.  
School Name / Degree  
MONTH 20XX - MONTH 20XX, LOCATION  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore.

## Canva Resume Templates (Paid)

If you have a subscription to [Canva Pro](#), you have the option of using hundreds of resume templates, included in the subscription. I've done many of the illustrations in this book [with Canva](#), and Canva is gaining more and more popularity, thanks to a good price-for-value monthly pricing.

### Rating for software developers:

From ★★★★★ to ★★★★☆

### The good:

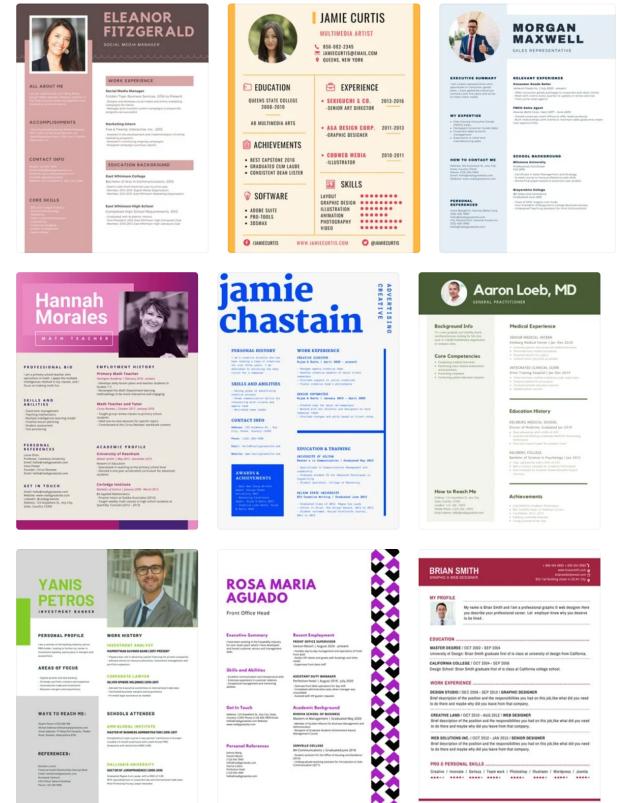
- Probably the largest selection of resume templates in one place.
- If you are a Canva subscriber, it's worth checking out the selection.

### The bad:

- The selection for templates is overwhelming and it's hard to narrow down to ones that work for tech.
- Most resume templates are optimized for looks over content.
- Many templates come with photos: something you'll want to avoid.

### Ways to improve it:

- There are decent templates with Canva: you just need to search more. Look for single-column ones that have clear formatting.



## Resume.io Software Engineer Template (Paid)

The [Software Developer Resume template](#) by resume.io is similar in style to the [Google Docs Serif resume](#) and it comes with the same issues as that one.

### Rating for software developers: ★★★☆☆

#### The good:

- A one-page resume.

#### The bad:

- The two-column layout limits the details you can share for your work experience.
- This resume breaks most guidelines in this guide. It does not have a Language and Technologies section—yet it has the unnecessary references section.
- The “Skills” and the blue line under them add noise for no reason.
- The “Skills” listed on this resume have nothing to do with skills hiring managers and recruiters look for at tech companies.
- The example resume copy is a poor example for a software engineer and seems to have been written by someone who has not worked in software.

#### Ways to improve it:

- I advise not to use this template for software engineering positions.

**Jim Ryan**  
Software Developer

#### Profile

Experienced Software Developer adept in bringing forth expertise in design, installation, testing and maintenance of software systems. Equipped with a diverse and promising skill-set. Proficient in various platforms, languages, and embedded systems. Experienced with the latest cutting edge development tools and procedures. Able to effectively self-manage during independent projects, as well as collaborate as part of a productive team.

#### Employment History

**Software Developer at Accruve Partners, New York**

September 2013 — September 2019

- Worked productively with Product Team to understand requirements and business specifications around Portfolio Management, Analytics and Risk.
- Effectively coded software changes and alterations based on specific design specifications.
- Worked to solve complex problems using the latest in Cloud, Mobile, and Web Technologies.
- Developed and presented findings and solutions to audiences including senior executives and stakeholders.

**Junior Software Developer at CyberCoders, New York**

June 2011 — August 2013

- Effectively addressed complex bugs.
- Implemented and updated application modules under the direction of Senior Software Developers.
- Successfully worked at an independent level, while also serving as an effective and enthusiastic collaborator.
- Performed automated testing tasks and developed complex features routinely.

#### Education

**Master of Science in Computer Science, The Massachusetts Institute of Technology, Cambridge**

August 2010 — August 2012

**Bachelor of Computer Science, Dartmouth, Hanover**

September 2009 — May 2010

#### References

**Dr. Carl Correnti from The Massachusetts Institute of Technology**

[ccorrenti@mit.edu](mailto:ccorrenti@mit.edu) • 617-228-3381

**Dr. Jennifer Peters from The Massachusetts Institute of Technology**

[jpeeters@mit.edu](mailto:jpeeters@mit.edu) • 617-283-2311

**Details**  
17 E 29th St, New York, 10016,  
United States, (917) 202-5269  
[jim.ryan@gmail.com](mailto:jim.ryan@gmail.com)

Nationality  
American

#### Skills

Advanced Analytical Thinking  
Skills

#### Programming

Software Logic

#### Software Troubleshooting

Knowledgeable in User Interface/  
User Experience

Adaptability

## Skills Points and the Resume.io Web Developer Template (Paid)

The [Web Developer Resume template](#) by resume.io is worth calling out for the pointed skills section—and why following skills scoring is not a great idea.

### Rating for software developers: ★★☆☆☆

#### The good:

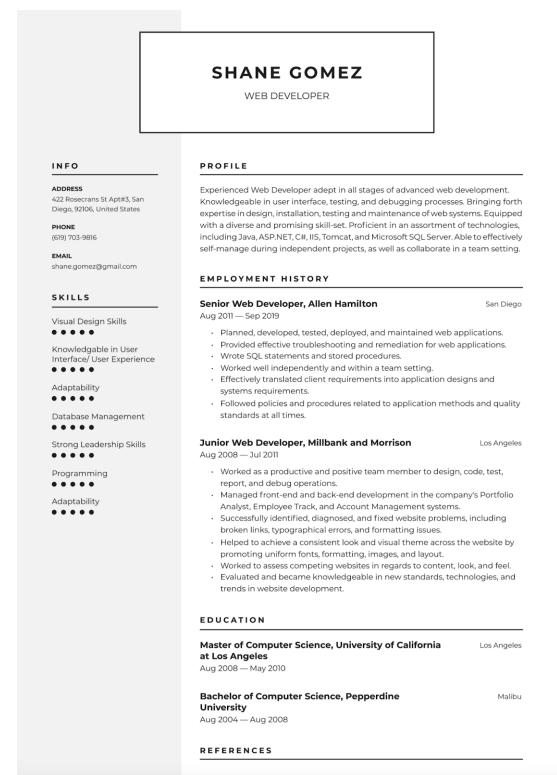
- A one-page resume.

#### The bad:

- The two-column layout limits the details you can share for your work experience.
- Note how the two columns are swapped from the previous resume.io template. You might ask yourself: why? It is a great question: there seems to be little consistency in resume template principles for many of the resume generator sites.

#### Ways to improve it:

- I advise to not use this template for software engineering positions.

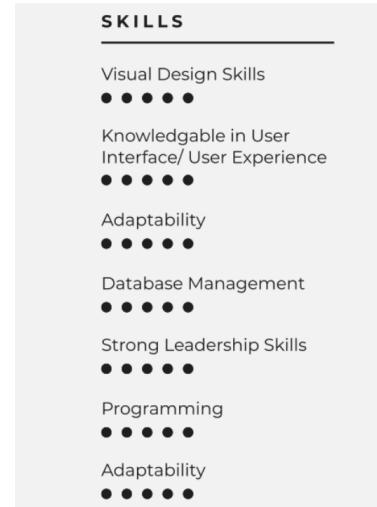


### On the points system

This resume template is an example of using the points system to “score” how strong you are in certain areas. However, here’s why this approach doesn’t work well:

- Anything you score yourself 3 points or lower out of 5 should likely not be on your resume. Why would you mention a technology in which you rate yourself at a low score?
- Anything mentioned as 4 or 5 stars is a given, without the star system as well. Recruiters and hiring managers will assume you are proficient with anything you list on your resume. Why else would you mention it?
- Your rating will likely be off anyway. I have seen people rate themselves as 5 stars on a language in which they seemed to not know things that I thought were basics. I’ve also had someone rate themselves 2 stars on a framework that they turned out to be solid at. People aren’t as good at rating themselves as they think they are—and experienced recruiters and hiring managers are very much aware of this.

Similar to how a profile photo won’t help your resume, star ratings can only work against you. Drop the ratings and keep all technologies relevant for the position that you are proficient with.



## VisualCV Standard Template (Paid)

The [VisualCV standard template](#) shares a lot of the good with the CareerCup resume. It's a solid alternative to that one—except for the fact that you have to pay to have the VisualCV branding removed.

### Rating for software developers: ★★★☆☆

#### The good:

- Clean format and nice use of colors.
- A resume format that works well for two-page resumes.
- Highlights your job titles while still making dates easy to read.
- Clear separation of sections.

#### The bad:

- Black-and-white, not much use of colors.
- When using the free service, a logo is added to the resume, which makes your resume look cheap.

#### Ways to improve it:

- Add a "Technologies and Languages" section or something similar on the first page.
- Make the summary short. You could also just drop the name "Summary".
- Drop the title "Software Engineer". It's obvious and takes up space.
- Consider jazzing up the headers and separators with colors.

## Andrew Anderson

Software Engineer

City State  
55444333  
hello@email.com  
[linkedin.com/in/yourna](#)

#### Summary

Software development engineer including four years of device driver development on all MS Windows operating systems. Expertise in the design and development of systems and software for high-speed data acquisition, process control and signal processing. Solid background in the technology of semiconductor devices, basic techniques of microelectronics, circuit design and analysis. Exceptional troubleshooting training and documentation skills.

#### Experience

##### Staff Software Engineer

2014 - Present

XYZ Company

Performed software design and development of hardware and the operation of specific sub-systems for the new Company SoftModem product.

- Ported SMS6 SoftModem to various hardware platforms (PCI based, AC97 based) for all Windows operating systems (Win9x/WinMe, Windows NT/2000, Windows XP).
- Supported test and application engineers in testing and debugging the modem software.
- Identified product specific hardware / software quality requirements to pass Microsoft WHQL certification.
- Provided support to client companies and end-users in the USA, Germany and Taiwan.
- Designed a C++ class library for Windows / Linux cross-platform kernel driver development.
- Participated in new product design and development.

##### Systems Engineer

2010 - 2014

ABC Company

Charged with the design and development of software for PC based data acquisition boards. Supported customers in the development of their software applications. Provided support to test department, field application engineers and hardware design engineers.

- Designed and developed device drivers for PCI-bus based data acquisition boards for Windows 95/98 and Windows NT.
- Created an application programmer interface (API) for various data acquisition boards. Implemented APIs in common function libraries (DLL's).
- Designed and wrote graphical user interfaces, application software and example programs.
- Developed and implemented LabVIEW virtual instrument libraries and Active-X controls.
- Designed and developed a function library for ABC Company C44 DSP processor board including the software interface and communication protocol between the DSP and host processor.
- Developed test procedures, test setup and software for final production tests as well as FCC and CE certification tests.
- Participated in the design, prototyping and testing of new analog and digital boards.

#### Education

##### Service Technician Certificate

2010 - 2011

Institute of Technology

# EnhanCV Software Engineer Resume (Paid)

I do not recommend the EnhanCV [Software Engineer resume](#) and am listing it here to showcase the dangers of resume sites that create resumes that job seekers find pretty. I listed this format as a cautionary example.

## Rating for software developers: ★★★★☆

### The good:

- Looks stunning—as a candidate, when you are looking at your own resume.

### The bad:

- Technical hiring managers and recruiters won't appreciate the format as it's hard to scan.
- Encourages adding a photo—which leads to bias.
- Some sections make little sense. Life Philosophy adds little content.
- A two-column structure that makes it hard to scan for recruiters, and discourages adding sufficient details for your experience.

### Ways to improve it:

- Do not add a photo.
- Power Skills are something you probably don't want on your CV: they're empty buzzwords.
- Only use this template if you can definitely fit everything on one page.
- Customize the right column and try to make good use of it. I'd remove "Life Philosophy", rename Toolbox to "Skills" and have a "Languages" and a "Technologies" section.

## ALEX JOHNSON

10 years challenging the concepts of finance, aiming to shape the future through technology



### IT PROJECTS

#### World Best Drive Buddy

4/2019 - 7/2019  
Full-stack Web application for sharing car journeys which allow for different people to join the same car ride/travel. The application enables both passengers and drivers to leave ratings, comments, feedback, etc.

Used technologies: Java, Gradle, Hibernate, Spring MVC, Spring Data JPA, JWT, MySQL, MariaDB, HTML, CSS, JS, Bootstrap

#### Worst Beers DataBase

4/2019 - 5/2019  
Full-stack application of a database of beers from all over the world, showing their details and average rating, enabling users to manage all the beers that they have drank and/or want to drink.

Being creative and seeking different approach we have crafted an application for the worst beers, not for the best.

Used technologies: Java, Spring MVC, Gradle, Hibernate, Spring Data JPA, MySQL, MariaDB, HTML, CSS, JavaScript

#### Kod Takoa

3/2019  
<https://github.com/angelov/Teamwork-1-telerik-academy-work-item-management-hub/master/KodTakoa20Presentation.pdf>  
Created a product with the brand "Kod Takoa", custom logo and product presentation. Console Application Developed as an OOP Teamwork.

Work Item Management System that provides a single view into all data, files, tasks, actions, collaborations, and history/incident (Bug), service request (Story), or process (Feedback).

### CORPORATE EXPERIENCE

#### Business Development Manager & IT Infrastructure Support

##### Amaze Corporate Events

8/2017 - 10/2019  
A leading team building and corporate events company in Bulgaria.  
Driving force behind generating new sales leads and contacts, boosted an increase in the customer database and revenue by 24% annually and upscaling additional services to existing clients leading a team of 3 people.  
I have developed an excel based business simulation for the financial sector.  
Maintained Google for Business cloud infrastructure, corporate websites, configuring 20 workstations and introduced new invoicing and project management systems.

### LIFE PHILOSOPHY

"Aut inveniam viam aut faciam."  
(I shall either find a way or make one)

Hannibal Barca

### TOOLBOX

#### TECHNOLOGIES

Java Spring MVC MySQL GIT  
Hibernate JavaScript HTML REST  
Thymeleaf CSS JWT security Gradle  
Bootstrap jQuery Postman

#### POWER SKILLS

Problem solving Assertiveness  
Creativity Critical thinking Teamwork  
Presentation / Public speaking  
Time management Emotional intelligence

### EDUCATION

#### Alpha Java Track

Telerik Academy

1/2019 - 7/2019 Sofia, Bulgaria

## EuroPass CV

Avoid this CV template in applying for tech jobs. The biggest sin of this format is encouraging adding all the details that lead to biases: photos, birth dates, gender, and so on. The Europass format is also *not* required to apply to jobs in Europe—contrary to what some people incorrectly assume.

**Every tech recruiter I've spoken with advised to not use this format** - and I advise the same. The Europass format was created by a working group within the European Commission. Though I could not confirm this officially, several recruiters mentioned hearing that there was no recruiter, HR specialist or job seeker in this committee.

### Rating for software developers: ★☆☆☆☆

#### The good:

- There is nothing good about the EuroPass format.

#### The bad:

- Encourages adding all the details that lead to biases: photos, birth dates, gender, and so on.
- Dated format that uses tables.
- Represents tech professions poorly
- Unnecessarily wastes space
- Adds details that are noise like spoken languages or driving licenses

#### Ways to improve it:

- Do not use this template for any positions—software engineering or not.



PERSONAL INFORMATION	Gergely Orosz																			
	 30 My Address, 1055AA London (Netherlands) +11122334455 hello@pragmaticengineer.com <a href="https://blog.pragmaticengineer.com/">https://blog.pragmaticengineer.com/</a>																			
JOB APPLIED FOR	Software developer																			
WORK EXPERIENCE	<hr/> <p>01/01/2019-Present Senior Software Engineer Uber ■ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua ■ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua</p> <p>2018-2019 Senior software engineer Skyscanner ■ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua ■ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua</p> <hr/>																			
EDUCATION AND TRAINING	<hr/> <p>2010-2014 BSc. Mathematics Budapest University of Technology</p> <hr/>																			
PERSONAL SKILLS	<hr/> <p>Foreign language(s)</p> <table border="1"><thead><tr><th></th><th>UNDERSTANDING</th><th>SPEAKING</th><th>WRITING</th></tr><tr><th></th><th>Listening</th><th>Reading</th><th>Spoken interaction</th><th>Spoken production</th></tr></thead><tbody><tr><td>English</td><td>B1</td><td>A1</td><td>A1</td><td>B1</td></tr><tr><td>Dutch</td><td>A1</td><td>A1</td><td>A1</td><td>A1</td></tr></tbody></table> <p>Levels: A1 and A2 Basic user - B1 and B2: Independent user - C1 and C2 Proficient user Common European Framework of Reference for Languages - Self-assessment grid</p> <p>Communication skills</p> <p>■ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua ■ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua</p> <hr/>		UNDERSTANDING	SPEAKING	WRITING		Listening	Reading	Spoken interaction	Spoken production	English	B1	A1	A1	B1	Dutch	A1	A1	A1	A1
	UNDERSTANDING	SPEAKING	WRITING																	
	Listening	Reading	Spoken interaction	Spoken production																
English	B1	A1	A1	B1																
Dutch	A1	A1	A1	A1																

---

## Chapter 13: Resume Improvement Examples

This section is a collection of real resumes that you can use as inspiration when writing yours. Remember that your resume should be about *your* experience, tailored for the *specific* position you are applying for.

I started writing this book after offering to review resumes for software developers who were looking for jobs. Having received too many requests for me to review in-depth, I started to send an early version of this book as feedback for people—this book was at 30 pages, so much shorter than it is today!

The below examples are real resumes that I received. They were often ones where the author was not receiving the type of response rate they were hoping for. You'll read the initial analysis of the resume, then see the improved version that the author ended up using as their "master" resume.

All examples are anonymized: names of people, companies, schools, locations and dates have been altered to not allow tracking back to the original author. The other features of the resume are as they were—giving you an idea of some typical areas that you could also address with your current resume.

**Resist copying anything one-for-one from these examples.** These resumes are someone else's experience. Use these to get inspiration, and see if your resume has similar improvement opportunities.

# **Software Engineer with 2 Years Experience**

The below resume is one for a person with a few years' experience, who intended to apply for Facebook's Rotational Engineering Program. Here is the description of the position, and the requirement details:

## **Facebook's Rotational Engineering Program**

Facebook is seeking talented full-stack Software Engineers who would like an opportunity to join our one-year Rotational Engineering Program, helping us build applications & systems that will scale the company and make the world more open and connected. The Program is designed to offer experienced engineers a one-year rotation through different Facebook engineering departments to develop engineering skills and gain meaningful experience for the rest of their career. At the end of the term, successful participants will be invited to apply for a full-time position at the company. Facebook is committed to increasing the diversity of representation among our Computer Scientists and Software Engineers.

We build products to connect the world, and this means we need teams that understand and reflect a broad range of experiences, backgrounds, identities, abilities and many other characteristics. The rotational engineering program is open to all qualified candidates. We strongly encourage candidates who are members of historically underrepresented groups and have non-traditional career paths to apply, including, but not limited to candidates from non-tech industries, candidates without Computer Science degrees who are self-taught, candidates starting second careers, re-entering the workforce, and those who have attended bootcamp-style programming courses.

## **Rotational Software Engineer Responsibilities**

- Full-stack product/systems development with a variety of languages such as C++, Java, PHP, JavaScript, etc.
- Work closely with product management and UX design teams to define and refine feature specifications.
- Design and develop front-end interfaces, underlying APIs, and backend systems across a number of programming languages with focus on JavaScript, React and PHP (Hack).
- Learn, learn and learn.

## **Minimum Qualifications**

- 2+ years experience building software solutions in a corporate or startup engineering environment, or equivalent, using JavaScript, PHP, Python, Ruby, C++ and/or Java.
- 2+ years experience developing web applications.

## **Preferred Qualifications**

- Bachelor's degree in Computer Science/Math/Engineering, or equivalent educational experience.
- Working knowledge of relational and non-relational databases and SQL.
- Working knowledge of API design and distributed backend systems.
- 2+ years' experience developing multi-tiered web apps.

# The original resume

## Edmond Smith

Mailing Address: TNC NT1, Cape Town  
E-Mail: edmondsmith88@example.com  
Website: <https://EdmondSmithDev.github.io/>

### TECH STACKS

- Programming Languages/Frameworks: PHP/Laravel, JavaScript/Express, Java/Kotlin for Android
- Database: Postgres, MySQL, MongoDB
- Infrastructure: AWS
- Testing Tools: , PHPUnit, Mockery
- API Specs: REST

### WORK EXPERIENCE

- Cloudless**, Backend Software Engineer(Remote) Sep 2018 – present
- Re-architected and maintaining a RESTful API using PHP/Laravel with third-party API integrations such as zoom-rooms to enrich customer experience.
- MennoMark**, Team Lead Sep 2017 – Aug 2018
- Managed a team of three developers on the development team.
  - Trained and on-board two new software developers on the software products(MennoMarker, MennoMarket).
  - Acted in leadership capacity and worked together with the rest of the development team to deliver quality software.
  - Built a mobile app(MennoInnova) for educating farmers enrolled in the Farmer Business Management programme. This was built with Kotlin, and SQLite Database.
  - Designed and implemented an API for interfacing the MennoInnova mobile application using PHP/Lumen, AWS(RDS, Elasticbeanstalk), PostMan.
- MennoMark**, Software Developer - Jan 2018 – Mar 2018
- Built an internal mobile app(MennoMini) for recording biometric data of poultry farmers in South Africa. This was built with Java, and SQLite Database.
  - Designed and implemented an API for interfacing the MennoMini mobile application using PHP/Lumen, AWS(RDS, Elasticbeanstalk), PostMan.
- South Business School**, BootCamp Trainer and Curriculum Developer Aug 2017
- Designed an android application development curriculum with focus on Java for the bootcamp.
  - Trained the participants with zero programming background to have a good understanding of Java and developed sample Android Application in a month.
- AdvancedLearning**, Technical Writer Mar 2017
- Updated previous technical articles to meet new features in JavaScript like classes.
  - Researched and wrote an article on building software with Angular which was ranked among the top ten articles for Angular in June 2018.
- Loud Fort**, Software Developer Intern Jun 2016 - Aug 2016
- Implemented API Designs in Golang programming language.
  - Documented APIs in Swagger(API Documentation and Designing tool).
  - Trained three interns on how to work with the Go Programming Language.
- The Trace Foundation**, Software Developer Intern May 2015 - Aug 2015
- Built an interface for a web application with HTML, CSS, Material Design and JavaScript(Angular).
  - Learned about backend development with the Go programming language.

---

## **VOLUNTEER WORK**

**[Facebook Developer Circles Cape Town](#)**, Developer Community Lead Aug 2016 - present

- Managing an online user group for software developers.
- Organizing meetups, workshops and hackathons for software developers in Cape Town.
- Building partnerships with local tech companies and communities like [JSKongress](#) to support the growth of the developer ecosystem.

**[Cape Town Tech Meetups](#)**, Technical Speaker

Aug 2017

- Gave a talk on building microservices with Java Spring Boot.

## **EDUCATION**

**[Cape Town University](#)** - Cape Town, South Africa Sep 2013 – Jun 2017

- Bachelor of Science in Computer Science

## **AWARDS AND ACHIEVEMENTS**

- [Developer Community Work Spotlight](#) - AdvancedLearning Dec 2018
- [MasterCard Foundation Scholar](#) - Cape Town University 2013 – 2017

## **SKILLS AND INTERESTS**

- Experience with architecture and design principles, building to consider maintainability, performance, security requirements, and impact.
- Experience working with software engineering collaboration tools like Git, Trello, Slack.
- Working knowledge of Object-Oriented Programming concepts i.e Classes and objects, Inheritance, Encapsulation, and Polymorphism
- Interested in developer community-building by hosting meetups, speaking at tech programs and technical writing.
- Experience working with Front End Technologies(HTML, CSS, JavaScript)
- Experience working with Front End Javascript Frameworks: Angular
- Experience with Front End Debugging Tools: Chrome DevTool

## **PERSONAL PROJECTS**

### **[bareGo](#)**

This project is usually for learning about software architecture and design using the PHP programming language. The concepts such as SOLID, Design patterns, etc, are applicable to other programming languages.

### **[RESTAPI with Node](#)**

Implementation of modern API design principles and concepts with Nodejs. Concepts include HATEOAS, Caching, Content Negotiation, etc.

### **[Technical Writing](#)**

I write about technical concepts that I learn about or find useful to the developer ecosystem on the Dev.to blogging platform.

## Analysis of the resume

Overall, this is not a bad resume—but there are many improvements that could make it a much stronger one for this position.

### The good

- **Key details on the first page.** Languages & technologies, as well as work experience, are listed on the first page.
- **Good amount of personal details.** This resume doesn't share too many unnecessary personal details.
- **A decent template and usage of bullet points.** The template helps with good usage of space, and bullet points make it easy to read the statements.

### Improvement areas

- **Overly generous spacing.** The margins on the sides of the resume were unusually large—reducing the space of what would fit on each of the pages. While good spacing can benefit resumes, there is no reason to have very large spacing on the top and the side.
- **Consistency.** Formatting is not consistent, and it brings a sloppy feel for the resume. Font sizes and alignment are different in different sections.
- **Drawing focus to the wrong thing.** This resume has many links—the person linked to all their previous companies. When glancing at the resume, the recruiter's attention is immediately drawn to the names of companies. However, it would be smarter to draw attention to the title of the engineer and the date. A different template could help with this, as would toning down link colors.
- **The impact of the work.** Within the work experience, there could be more specific impact mentioned about the person's work. There are some specifics, but there are also a lot of phrases like “Built a mobile app”, without explaining what was impactful, challenging or interesting about it.
- **Linking to company websites.** The links to company websites drew attention but brought little value. If a recruiter or hiring manager wants to look up the company, they can do so without a link, just by searching for the company name. Besides, the company landing page does not convey much relevant information.
- **Splitting out a 3-month position at MennoMark.** This person got promoted after a few months at MennoMark. They decided to split this 3-month-long position out into separate sections. Because of how fast this promotion was, they could just ignore this short position, not mentioning it.

# The refactored resume

## Edmond Smith

Cape Town, South Africa

edmondsmith88@example.com

[Website](#)

### TECHNOLOGIES AND LANGUAGES

- PHP, JavaScript, Java, Kotlin, Android, Angular, Laravel, Express.js, Ktor
- MySQL, MongoDB, AWS, Git
- Data structures and algorithms, API design, engineering best practices, unit testing

### WORK EXPERIENCE

**Backend Software Engineer**      **Cloudless**      **Sep 2018 - present**

Remote

- Improved customer conversion by 30% of the Whack co-working platform by implementing recurring payments and integrating Zoom rooms.
- Re-architected the restful API powering the mobile client for Whack, using PHP and Laravel.

**Development Team Lead**      **MennoMark**      **Jan 2017 - Aug 2018**

- Led a team of 3 engineers to ship multiple apps and services.
- Built an app for educating cocoa farmers enrolled in the Farmer Business School programme, using Kotlin and Room database.
- Designed and built a mobile app to record biometric data of poultry farmers in South Africa, using Java, and SQLite database.
- Designed and implemented APIs that powered our 10 mobile applications using PHP, Lumen, Laravel, AWS RDS, AWS Elasticbeanstalk and PostMan.
- Onboarded and trained 6 new employees on the company's development and product stack.

**Developer and Trainer**      **South Business School**      **Summer 2017**

- Designed a month-long Android application development bootcamp curriculum with 30 modules and a strong Java focus.
- Trained 40 participants from zero programming background to having a good working knowledge of Java, and being able to build Android applications in a month.

**Guest Author**      **AdvancedLearning**      **Summer 2017**

- Researched and wrote an article on building software with Angular Material UI which was ranked among the top ten articles for Angular in June 2018.
- Reworked technical articles, updating them to reflect ES6, ES7 and ES8 Javascript features

**Software Developer Intern**      **Loud Fort**      **Summer 2017**

- Built an internal web tool to manage leads more efficiently, used by 30 sales people. Built using JavaScript, Angular, Go, HTML and CSS.

### EDUCATION

**Cape Town University**      **2013 - 2017**

- Bachelor of Science in Computer Science
- Received the MasterCard Foundation Scholar, awarded to 10 out of 300 applicants.

---

## PROJECTS AND VOLUNTEER WORK

---

- **Community Lead** at [Facebook Developer Circles](#). I built up a local group of 500 developers, organizing 30+ events and hackathons. Featured on the [Community Storytellers' Series](#).
- **BareGo** - A minimal and progressive PHP framework for building efficient and scalable server-side applications. See the source [on Github](#).
- **REST API with Node** - Modern API design principles and concepts with Nodejs. Concepts include HATEOAS, Caching and Content Negotiation. See the source [on Github](#).
- **Technical blogging** on [Dev.to](#) and as guest publisher. Popular articles include a piece [on Angular Material](#) and one [on Clean Code](#).

## Analysis of the refactored resume

The refactored resume addressed most of the improvements we previously mentioned. A few things to call out with this improved resume are:

- **Tailored for the position the person is applying for.** For example, the position mentioned languages like C++, Java, PHP, JavaScript etc. As the person is hands-on with PHP and JavaScript, the resume lists these first.
- **Impact and what this person actively did comes across more clearly.** The updated resume uses active language and showcases specific examples of impact. Note how this is exactly the same person, talking about exactly the same work they have done. In the updated version, however, they have gone deeper to express how their work was important beyond just coding.
- **More concise wording, with just as much details.** The two resumes convey roughly the same amount of information. However, in the second one, much of the irrelevant or repetitive information has been cut.
- **More deliberate guiding of the reader's focus.** In the original resume, links were scattered. Take the technical blogging section: in the original resume, this was just a link to the Dev.to page of this engineer. Now, instead of linking to a blog, this person calls out two articles they'd like the hiring manager to glance at. This is a much more focused approach.

## Ways this resume could be made even better:

- **A short summary section** is something I'd consider adding, especially if there's information this candidate can reflect on for the job description. For example, the job description says: "*We strongly encourage candidates who are members of historically underrepresented groups and have non-traditional career paths to apply, including, but not limited to candidates from non-tech industries, candidates without Computer Science degrees who are self-taught, candidates starting second careers to apply.*" If some of this applies to the candidate, I'd encourage them to reflect on this in this section.
- **Tailor the resume to other positions**, over using this one as a "one-size-fits-all" one.

---

## **Machine Learning Engineer with 5 Years Experience**

This resume is from a software engineer in Europe who specialized in machine learning. They were applying to companies but rarely heard back. Their original resume was a 3-pager one.

After refactoring the resume to a 1-pager one below, they saw a lot more leads, including from companies that they did not hear back from before. The person ended up landing a job at one of the companies they were targeting.

### **The original resume**

# PAT THOMPSON

MACHINE LEARNING ENGINEER @  
LASTMINUTE.COM LTD

Location: Vienna, Austria  
Github: [https://github.com/pat\\_t](https://github.com/pat_t)  
LinkedIn: [https://www.linkedin.com/in/pat\\_t](https://www.linkedin.com/in/pat_t)  
Email: pat\_t@icloud.com  
Site: [https://pat\\_t.dev/](https://pat_t.dev/)

## EXPERIENCE

Machine Learning Engineer  
Lastminute.com, Vienna, Austria  
2018–current –Full-time

Backend Python Developer  
CARRS,Vienna, Austria  
2016 –2018 –Full-time

Software Developer  
Lending Limited LP, San Mateo, CA, US  
2015–2016–Full-time (remote)

Machine Learning Developer  
Inno Systems, San Mateo, CA, US  
2015–2015–Contract (remote)

Founder & Software Engineer  
Future Co, Sao Paulo, Brazil  
2014 –2015 –Full-time

## FRAMEWORKS & LIBRARIES

### Machine Learning:

Tensorflow, Spacy, NLTK,  
Sklearn, OpenCV

### Web:

Django, Flask, Vuejs, Nodejs,  
Phoenix

### Data Processing:

Spark, Pandas, Numpy

### Other:

Selenium, beautifulsoup4, Click

## EDUCATION

B. S. of Computer Science •2015  
University of São Paulo, Brazil

## PROGRAMMING LANGUAGES

Python	● ● ● ● ●
Javascript	● ● ● ● ●
Elixir	● ● ● ● ●
Go	● ● ● ● ●
Kotlin	● ● ● ● ●

## CLOUD

### AWS:

EC2, RDS, EMR, ECS, SageMaker,  
S3

## TOOLS

Docker, Terraform, Packer,  
PostgreSQL, MySQL, MongoDB  
Snowflake, Airflow,Jenkins,  
Kubernetes, Nvidia, CUDA

## DESIGN PATTERNS

REST, MVC, Microservices, TDD,  
Server-Side-Rendering

## DETAILED EXPERIENCE

### Machine Learning Engineer

Lastminute.com, Vienna, Austria

2018–current –Full-time

Lastminute.com is one of the largest websites in the world that runs best-in-market flash sales of hotels and holidays. It operates in 21 countries and offers its members exclusive rates of up to 60% off!

As a Machine Learning Software Engineer I bring in my technical development expertise and knowledge to Data Science team. My main purpose is to support the business by utilizing data science solutions and building production-ready data products.

Things I work on:

- Implementing and optimising data pipelines, monitoring and alerting, etc to ensure data flows correctly into models;
- Help build up tooling and infrastructure for Data Science team for model deployment, management and performance monitoring;
- Facilitate integration of work from Data Science team into production environments by scoping and defining interfaces with production environment;
- Provide technical best practice to team;
- Set up standardised libraries of tooling for data manipulation, models, etc to improve speed of development for data scientists;
- Liaise with key stakeholders Engineering and Data teams
- Keep myself updated on new developments in your field of expertise.

Stack: Python, Flask, Docker, Terraform, Git, AWS, Tensorflow, NLTK, Spark, RESTful APIs, Snowflake, Postgres, Airflow, CUDA+Nvidia (for GPU intensive processes).

### Backend Python Developer

CARRS, Vienna, Austria

2016 – 2018 –Full-time

CARRS is a leading car trading platform. By connecting buyers and sellers across through technology, CARRS enables private customers as well as professional partners to trade seamlessly.

Here I was responsible for:

- Build, test and maintain innovative tools to automate the company's internal processes within the marketing department;
- Implement high performance and complex algorithm-based solution;
- Perform data analysis on databases using machine learning algorithms;
- Implement algorithms to perform text classification and sentiment analysis.

Stack: Python, Flask, RESTful APIs, SQL, ORM libs, Docker, AWS, Git, Microservices, Keras/SKLearn (Machine Learning), NLTK/Spacy (Sentiment Analysis).

## **Software Developer**

Lending Limited, San Mateo, CA, US  
2015–2016–Full-time (remote)

Lending Limited is a private direct lender to fix-and-flip investors and the first fund with more than \$100M in loans in a single year — There I was responsible for developing (in a test driven approach) new features and maintaining the company's internal micro-services and web apps used to handle all of its business processes.

Stack: Python, Pyramid, Django, HTML/CSS/JavaScript, MongoDB, AWS, Git, Mercurial, Microservices, Docker, Go.

## **Machine Learning Developer**

Inno Systems, San Mateo, CA, US  
2015–2015–Contract (remote)

Inno Systems' primary mission is to provide practical solutions for real-world business problems using deep learning and machine learning solutions. There I was responsible for developing Machine Learning algorithms exploring several of its subfields concepts such as Data Mining, Text Classification but mainly Computer Vision in a Game Theory project.

Stack: Python, C++, OpenCV, Sklearn, Pandas, Numpy, Git, AWS, NTLK.

## **Founder & Software Engineer**

Future Co, Sao Paulo, Brazil  
2014–2015 –Full-time

Future Co. was a software factory type of company which I founded during my last year in university. Our goal was to provide software solutions to small local business in the city where I studied. Besides managing the business as whole, I also acted as a Software Engineer designing, developing and testing our applications for a diverse client base.

Stack: Python, Java, Django, HTML5/CSS3/JavaScript, Nginx, JSON, Postgresql, MySQL, AWS, DigitalOcean, Git.

## **ADDITIONAL INFORMATION**

### **LANGUAGES**

Portuguese	● ● ● ● ●
English	● ● ● ● ●
Spanish	● ● ● ● ●
Italian	● ● ● ● ●
German	● ● ● ● ●

### **TALKS**

Scanning the Web with Python.
2014-II Python Beach, Brazil
Intro to Python multi-platform framework for Apps Development.
2013-I Python Beach, Brazil

---

## Analysis of the resume

The resume is an instant turn-off. The template is unnecessarily hard to read. The first page feels like it was written just to have buzzwords on it. Also, it immediately tells the story of someone who is either not a good developer or just overly modest. Why would the resume start with this person rating themselves as 4/5 on Python, 3/5 on Javascript?

### The good

- **The second and third page** shows depth in experience. Unfortunately, most recruiters and hiring managers likely never got here, and moved this resume to the “reject” pile after scanning the first page.

### Improvement areas

- **Use an easy to read template.** The current template makes it hard to scan the resume contents. The black headers draw attention needlessly.
- **Do not use the skills self-rating.** As discussed in the [Unnecessary Details section](#), self-rating skills proficiency can only work against you on a resume. This can be seen here, in action. As a hiring manager, I could assume this person is not a good developer, based on their own rating. If they rate their Javascript as 3/5, some hiring managers could assume they are not quite proficient—which is not at all what this person wanted to convey.
- **Talk about impact and results**, not what you were told to do. In a new field like machine learning, most companies will look for people who take the initiative and get results.
- **Convey the right type of information on the first page.** This person has a lot to build on: so do this. They should showcase the impactful work they've done, and their skills.
- **Talk more about themselves, not their companies.** For every position listed, this person writes nearly as much about the company as they do for their achievements. Drop the part on the company: they should focus on their results.
- **Use bullet points** for easier reading. Much of the resume has paragraphs that list different pieces of information. For example, the first sentence talks about the company and the second one about their achievements. After reading the first sentence, the person scanning the resume would skip to the next paragraph, assuming that a paragraph talks about the same topic.

## The refactored resume

# Pat Thompson

pat\_t.dev

github.com/pat\_t

linkedin.com/in/pat\_t

pat\_t@icloud.com

Vienna, Austria

## EMPLOYMENT

<b>Machine Learning Engineer</b>	<b>Lastminute.com -Vienna Austria</b>	<b>2018 -current</b>
• Created infrastructure needed for development, staging, and production within the Data Science pipeline.		
• Reduced 50% of costs on cloud hosting by building an in-housescaling tool to manage cloud resources.		
• Improved the speed of Spark clustering algorithmswhilst reducing the cost by building the infrastructure to use AWS EMR (Elastic Map Reduce).		
• Built and exposedAPIs for model prediction in real-time.		
• Created custom Tensorflow Docker image to use it in AWS Sagemaker.		
• Created continuous integration and deployment pipeline.		
<b>Backend Engineer</b>	<b>CARRS -Vienna, Austria</b>	<b>2016 -2018</b>
• Built software tools to interact and automate several different online marketing services suchas Google Ads, Facebook, and Instagram.		
• Developed a sentiment analysis model to evaluate and tag customer feedback, saving hours of manual labor.		
• Built and exposedAPIs to serve predictions.		
• Created a full-stackweb application for internal stakeholders.		
• Built infrastructure for deployment on AWS usingTerraform.		
<b>Software Engineer</b>	<b>Lending Limited -San Mateo, CA, US (Remote)</b>	<b>2015 - 2016</b>
• Built several microservices for the main product.		
• Developed automation tool for document generation.		
• Created a full-stackweb application for the company's blog.		
<b>Machine Learning Engineer</b>	<b>Inno Systems -San Mateo, CA, US (Remote)</b>	<b>2015 -2015</b>
(Contract)		
• Designed and developed a Computer Vision algorithm to detect dices and player moves in a video stream.		
• Created infrastructure for local development using tools such as Docker.		
<b>Founder &amp; Software Engineer</b>	<b>Future Co. -Sao Paulo, Brazil</b>	<b>2014 -2015</b>
• Founded the software factory company during the last year of university.		
• Designed, developed, and productionised software for several customers.		
• Led a team of 4 people: 3 developers and 1 product designer.		
• Interacted and gathered feedback from clients and stakeholders.		

## EDUCATION

<b>Sao Paulo, Brazil</b>	<b>University of Sao Paulo</b>	<b>2011 -2015</b>
B.S. of Computer Science		

## TECHNOLOGIES

<b>Programming Languages</b>	<b>Frameworks &amp; Libraries</b>	<b>Tools</b>
Python, Javascript, Elixir, Go, SQL	Django, Flask, Vuejs, Nodejs, Phoenix, Tensorflow, Spacy, NLTK, Sklearn, OpenCV, Selenium, beautifulsoup4, Click, Spark, Pandas, Numpy.	AWS, Docker, Terraform, Packer, PostgreSQL, MySQL, MongoDB Snowflake, Airflow, Jenkins, Kubernetes, Nvidia, CUDA.

---

## Analysis of the refactored resume

The difference between the original and the refactored resume is striking. The resume is down to one page, is a lot easier to read and conveys the applicant's experience much better. The person applied much of the advice in this book.

They also saw a lot more leads, including from companies that they did not hear back from before. The person ended up landing a job at one of these companies.

Still, there are a few further areas I'd suggest looking at:

- **Mention specific technologies** they worked with, in the work experience. E.g. instead of "*Created a full-stack web application for internal stakeholders*", they could say "*Created a full-stack web application for internal stakeholders using Vue.js Django, deploying it on AWS.*"
- **Use numbers** when describing the work experience. E.g. instead of "*Created a full-stack web application for internal stakeholders*", list specifics like "*Designed, implemented and deployed 4 microservices for the main product. These services serve 100 RPS.*"

---

## **Backend Developer with 6 Years Experience**

This example is from a backend engineer who spent a few years working at a company building payroll software. They originally intended to use their resume to apply to similar, but larger companies.

With their original resume, they were seeing far less response than they were hoping for. With the improvements, they noticed a considerably higher rate of callbacks, and ended up accepting an offer for a local startup for a senior engineer position.

### **The original resume**

# Nikita Pratley

Phone: +1-234-567-8910

Software Developer

Email: nikita.pratley@gmail.com

LinkedIn: in/nikitapratley

## Technical Skills

Java, SQL, Oracle Database, PL/SQL, SQL Server, Microsoft Azure DevOps (TFS), SQL Serve, Web Technologies and Architectures (REST, JSON, XML, SOAP, HTTP), Linux, JavaScript, Python, Jira, Git, SVN, Docker.

## Work Experience

### Software Engineer – R&D

PaySolutions - Vancouver, Canada, February 2019 – Present

- Led development of two separate **third party interface development projects** for two of our biggest clients with Workday and ADP, 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%.
- Designed and developed the new **W4 interfaces** for our existing US-based clients to provide **seamless tax withholding experience** with their payroll (Workday, UltiPro, ADP, Kronos, WFR, Infinity HR) system.
- Redesigned the **SAP Concur interface** for our biggest client that benefitted them by **performance improvement of more than 300%**.
- Trained Developers and peer reviewed code to maintain company and industry standard.
- Technologies: Java, Oracle Database, PL/SQL, Postman, T-SQL, MS SQL Server, Azure DevOps (TFS), SQL Server Studio Management, Git.

### Software Engineer – Customer Success

PaySolutions - Vancouver, Canada, June 2017 – February 2019

- Created troubleshooting tools that improved 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.
- Developed **the data restriction feature** of the compensation reporting tool which enabled clients to generate their own customized reports.
- Technologies/ Frameworks: PL/SQL Developer, SQL, JavaScript, Oracle database, Git, SVN, HTML, CSS, Jira.

### Intern

Nokia, Helsinki, Finland, April 2016 – May 2017

- Monitored and instantaneously **categorized user's behavior** using different clustering algorithms (Gaussian Mixture Models (GMMs), K-Means).
- Identified the direction and the speed of change of customers from one cluster to another to create and hypothesize **behavior trajectory** for predicting customer behavioral changes.
- Technologies: R, R Studio.

### Software Engineer

GraphTL, Bangalore, India, November 2012 – May 2014

- Automated the **testing of various web app** using SoapUI and TestComplete.
- Technologies: JavaScript, Perl, SoapUI, Jira, TestLink.

## Education and Certifications

- **M.Sc. Computer Science**, LMU University of Munich, Munich, Germany.
- **B.Sc. Computer Science**, BIT Bangalore Institute of Technology, Bangalore.

## Analysis of the resume

This resume is a decent one to start with—however, it can be further improved.

### The good

- **Technical skills on the top.** This person was aiming to apply for companies that hired for specific tech stacks. In this case, starting with relevant technologies is a good strategy.
- **A one-page resume.** This person had around five-six years' experience, and they managed to fit the key parts onto one page.
- **Mentioning results and numbers.** The resume had specifics, like the percentage of reducing the client ticket support rate or performance improvements.
- **Good amount of personal details.** This resume doesn't share too many unnecessary personal details.
- **A decent template and usage of bullet points.** The template helps with good usage of space, and bullet points make it easy to read the statements.

### Improvement areas

- **Missing location** on the resume. When no address is listed, recruiters might assume the person is out of the country. This was not the case with this person; they were based in Vancouver, Canada. So why not list it?
- **Technologies dumped one after the other.** While it's good to list all technologies this person is proficient with, the formatting and the order could be better. Focus on languages and frameworks that are needed for the position, or that this person wants to call out specifically.
- **Too much bolding.** Much of the bolding is distracting, and some of it makes little sense. Why are all the technical skills in bold? Why are some of the results bolded out? Why are "W4 forms" bolded? Though the person writing the resume probably wanted to be helpful, they draw attention to the wrong areas.
- **Could use more specifics.** Though there are specifics for some bullet points, others could use more details. Take the statement "*Created troubleshooting tools that improved support team's efficiency by more than 30%*". What languages or frameworks did this person use for the tools? Did they come up with an idea? Did they lead a team? How many customers used the tools?
- **Grammar.** Some of the sentences don't read naturally: using a grammar checker would have surfaced these. Taking the previous example: "*Created troubleshooting tools that improved **the** support team's efficiency by more than 30%*".
- **Education dates are missing.** As a hiring manager and a recruiter, not seeing education dates is confusing. Did the person graduate? If so, why not add the dates? If they did not, why don't they mention that they dropped out?
- **Dates could be better placed.** Moving the dates to the right side would make them much easier to scan on the first glance.
- **Docx format.** The original resume was a docx. Using PDFs is a much better choice.

# The refactored resume

Vancouver, Canada  
[LinkedIn](#)

**Nikita Pratley**

(234) 567 8910  
nikita.pratley@gmail.com

## Technical Skills

- Java, JavaScript, Python, SQL
- Oracle Database, PL/SQL, SQL Server, Microsoft Azure, SQL Server, Git, Docker, REST
- DevOps, web technologies and architectures, data structures and algorithms

## Work Experience

Software Engineer	PaySolutions	Feb 2019 - Present
R&D team	Vancouver, Canada	
<ul style="list-style-type: none"><li>• Reduced operation costs by 45% for a client by developing third party interfaces using Java, Workday and ADP.</li><li>• Reduced support tickets by 80% for a large client by automating error handling on their restful API</li><li>• Improved workflow efficiency by 300% for a client after I redesigned the client's SAP Concur interface</li><li>• Shipped a next-generation tax withholding experience, integrating with Workday and several other payroll systems.</li><li>• Trained 5 developers on coding practices and contributed to the company-wide code review guide.</li></ul>		
Software Engineer	PaySolutions	2017 - Feb 2019
Customer Success team	Vancouver, Canada	
<ul style="list-style-type: none"><li>• Led a team of 3 to ship a compensation reporting tool, rolling it out to 15 different clients.</li><li>• Reduced the time to generate data transaction files from 5 minutes to a few seconds by designing, implementing and rolling out the mass transactions uploader platform.</li><li>• Improved the support team's throughput by 30% by building troubleshooting tools using Javascript.</li></ul>		
Intern	Nokia	2016 - 2017
	Helsinki, Finland	
<ul style="list-style-type: none"><li>• Built a real-time user behaviour monitoring system that used several different clustering algorithms like (Gaussian Mixture Models (GMMs) and K-Means.</li><li>• Predicted customer behavioral changes by identifying the direction and the speed of change of customers from one cluster to another. I created and hypothesized behavior trajectory using R and R Studio.</li></ul>		
Software Engineer	GraphTL	2012 - 2014
	Bangalore, India	
<ul style="list-style-type: none"><li>• Reduced regressions by 60% after I automated the testing of our web apps using SoapUI and TestComplete, building tests with Javascript and Perl.</li></ul>		

## Education and Certifications

- |  |             |
|--|-------------|
| <ul style="list-style-type: none"><li>• <b>M.Sc. Computer Science</b>, LMU University of Munich, Munich, Germany.</li><li>• <b>B.Sc. Computer Science</b>, BIT Bangalore Institute of Technology, Bangalore.</li></ul> | 2010 - 2012 |
|  | 2007 - 2010 |

---

## Analysis of the refactored resume

The refactored resume addressed all the above improvement areas, and the person ended up with more calls and an offer with a local startup. The main improvements from the original resume were:

- **Mentioning their location** as Vancouver, Canada and not omitting it, like before.
- **Languages and technologies listed more strategically** and tailored for the job they are applying for. This person was going after mostly Java-based positions. They listed programming languages first, then the technologies relevant for the job that they were proficient in.
- **An easier to scan template** with dates and positions standing out.
- **Starting with the impact of their work** in their work experiences, instead of burying this information deeper.
- **Giving more specifics** on their work experience. For example, they mentioned leading a small team, which they previously left off.
- **Dropping months from older dates**—from four or more years back. It makes the dates easier to scan. This person didn't have a gap in their employment since 2016, and this can make scanning their resume a bit easier.
- **Adding education dates** to add more context on how long they have been doing software development.

---

## **Remote Software Engineer With 8 Years Experience**

The below resume is a 2-page CV for a senior software engineer based in Europe. This person was primarily targeting remote positions.

The candidate behind this resume struggled to get responses from international companies, even though they were a senior candidate. Following the refactor, their response rate increased dramatically, and they ended up choosing one of the multiple offers they secured.

### **The original resume**



# DAN MITCCH

## SOFTWARE ENGINEER

📞 +32-20-222-2222  
✉️ DAN.MITCCH@GMAIL.COM  
📍 WARSAW, POLAND

### OPEN FOR REMOTE POSITIONS

#### EDUCATION

WARSAW UNIVERSITY OF  
TECHNOLOGY  
2007-2011

SYSTEMS DEVELOPMENT



#### FIND ME

- LINKEDIN
- GITHUB
- HACKERRANK



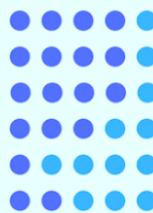
#### EXPERIENCE

- ERICSSON COMMUNICATIONS 2012-2013  
SOFTWARE DEVELOPER  
TEST RESPONSIBLE
- TTL LLC 2013-2016  
SOFTWARE DEVELOPER  
TECHNICAL LEADER
- SUMO LOGIC 2016-2018  
ENGINEER OF ENGINE TEAM
- DRIVE LLC 2019-2020  
DEVELOPER, ARCH-DEV TEAM
- FREELANCER 2020-  
—



#### SKILLS

C++  
QT  
PYTHON  
AUTOMATION  
NODE.JS  
CAD/CAM



#### OTHERSKILLS

- ISTQB-CERTIFIED TESTER
- LEADERSHIP
- SOURCE CONTROL, ETC.
- CAT. B DRIVING LICENSE



## DETAILED EXPERIENCE

<b>FREELANCER (HOMESHOP TECHNOLOGIES LLC)</b>	2020-
<ul style="list-style-type: none"><li>• WORKING AS A FREELANCER IN THE E-COMMERCE INDUSTRY</li><li>• MOSTLY DEVELOPING DATAPROCESSING MICROSERVICES RUNNING IN THE CLOUD</li><li>• DEVELOPMENT OF A LARGER B2B SOFTWARE USING PYTHON</li></ul>	
<b>DRIVE LLC</b>	2019-2020
<ul style="list-style-type: none"><li>• DEVELOPER OF ARCH-DEV TEAM</li><li>• CUSTOMIZING EXISTING SOLUTIONS TO SPECIALIZED PLATFORMS (NVIDIA DRIVEPX)</li><li>• BUILD SCRIPT MAINTENANCE(JENKINS, CMAKE, PYTHON, BASH)</li><li>• PERFORMANCE AND QUALITY IMPROVEMENTS (C++)</li><li>• ARCHITECTURE DESIGN</li></ul>	
<b>SUMO LOGIC</b>	2016-2018
<ul style="list-style-type: none"><li>• DEVELOPMENT OF THE MULTIPARTITION GRAPHICAL ENGINE (C++11 AND 14)</li><li>• UNIT TESTING (GTEST)</li><li>• BUILD SCRIPT MAINTENANCE(JENKINS, CMAKE, PYTHON, BASH)</li><li>• ENGINE INTEGRATION TO PLATFORM APPLICATIONS (TYPESCRIPT, OBJECTIVE-C, JAVA)</li><li>• GIVING A HELPING HAND IN QA PROCESS DESIGN</li></ul>	
<b>TTL LLC</b>	2013-2016
<ul style="list-style-type: none"><li>• DEVELOPMENT OF LINE-FIT NAVIGATION SOLUTIONS (C++, LUA, PLATFORMS: WINCE AND ANDROID)</li><li>• TECHNICAL LEADERSHIP OF DEVELOPMENT PROJECTS (COMMUNICATION WITH CUSTOMERS AND INTERNAL TEAMS)</li><li>• CODE BRANCH AND BUILD ENVIRONMENT MANAGEMENT(SVN, TEAMCITY)</li></ul>	
<b>ERICSSON COMMUNICATIONS</b>	2012-2013
<ul style="list-style-type: none"><li>• WEB DEVELOPMENT AS INTERN (PHP, JAVASCRIPT, JQUERY)</li><li>• TEST AND BUILD ENVIRONMENT AUTOMATION AS TEST RESPONSIBLE (HIT, BASH, QT, SVN AND GIT)</li><li>• FAULT CORRECTION AND GENERAL SOFTWARE DEVELOPMENT (TNSDL AND C)</li><li>• TOOL DEVELOPMENT (BASH, QT, C++)</li></ul>	

---

## Analysis of the resume

I was not surprised to see that this person barely received calls from recruiters. This resume did an excellent job in breaking several unwritten CV "rules" and downplaying the candidate's experience.

### The good

- **Not much good to say** about this resume, beyond the fact that the person behind the resume has a *lot* of experience to build on. Let's jump instead to the biggest improvement areas.

### Improvement areas

- **Confusing layout** that starts with education. Most recruiters would stop reading after the first page - however, the second page contains most of their "proper" experience.
- **Photo** that introduces bias. Just remove it.
- **Irrelevant skills listed.** Leadership? Cat-B driving license for a remote developer job? Remove these.
- **Skills with 1-5 points** make it seem like this person is a poor engineer. They rate themselves 4 stars at most, and one star on Node.JS. Talking to this person, they were an expert or proficient on most technologies. I advised them to remove the rating altogether.
- **Reverse chronological work experience.** On the first page, the oldest experience is listed first. This is both confusing and shows this person is unaware of resume basics.
- **Only talking about "what" they did, not the results.** The resume does not use active language, and does not do a good job describing the results this person delivered.

## The refactored resume

**DAN MITCCH**  
**MOBILE:** +3220 2222222  
**E-MAIL:** dan.micch@gmail.com

[LINKEDIN](#)

[GITHUB](#)

## Skills

### Languages and technologies

- Expert in Python and C++ and a wide variety of frameworks for these languages, for example: Flask, Selenium, Numpy, Qt5, Boost, POCO
- Experienced with a number of CI systems, testing frameworks and designing/running services in a cloud-based architecture
- Familiar with designing and using multiple database systems, for example SQL, noSQL databases
- Familiar with a number of other programming languages, for example: JavaScript, Java, TypeScript and C#
- Proficient in API and architecture design, experienced in technical and team leadership
- ISTQB Foundations certified software tester, experienced in TDD
- English (C1 level certification by TELC Hungary)

### Education

Engineer of Computer Sciences, BSC at Warsaw University of Technology and Economics, graduated in 2011.

### Work experience

<b>Freelancer</b>	<b>Homeshop Technologies</b>	<b>2020 spring –</b>
• Creating backend microservices for customers operating in the e-commerce market to implement automated business solutions		
• Up to 10 times increase in store throughput in some cases, with up to 50% decrease in manual labor requirements		
• Technologies used: Python (with Flask, Selenium and Numpy), C++ (with Qt framework), Linux and Cloud architecture development, JavaScript, git, CMake and qmake		
<b>Developer, Arch-DevTeam</b>	<b>Drive LLC</b>	<b>2019 summer – 2020 spring</b>
• Architecture design and software development to create production/demoready solutions from research findings for self-driving vehicles		
• Increased performance of stereo-camera lane detector solution by 100%, enabling it to run on an nVidia DrivePX2 or Jetson embedded hardware with acceptable reaction time in less than 2 weeks		
• Technologies used: C++(14), Qt, Python, CMake, nVidia SDKs, git		
<b>Sabbatical break</b>		<b>2018 winter – 2019 summer</b>
• Studying Python through smaller projects and learning how to operate and maintain 3-axis CNC mills		
<b>Engineer of Engine Team</b>	<b>Sumo Logic</b>	<b>2016 – 2018 winter</b>
• Application developer responsible for developing new features and maintaining the multi-platform graphics engine of the company		
• Completely refactored the network interface of the engine on all platforms to move resource procession to the cloud, resulting in a 20–100% decrease in load times		
• Technologies used: C++(11 and 14), GTest, Python, CMake, Jenkins, Java, Objective-C, TypeScript and JavaScript, git, emscripten, OpenGL (ES 2.0), webGL		
<b>Technical Leader (senior)</b>	<b>TTL LLC.</b>	<b>2015 – 2016</b>
• Technical leadership of multiple line-fit navigation software solution projects		
• Successfully negotiated and managed the technical and scheduling details between clients, 3rd-party		

**DAN MITCCH**  
**MOBILE:** +3220 2222222  
**E-MAIL:** dan.micch@gmail.com

[LINKEDIN](#)

[GITHUB](#)

vendors and other departments of the company

- Delivered the first SCOOP integration with a GUI

**Technical Leader**                           **TTL LLC.**                           **2014 – 2015**

- Technical leadership of a line-fit navigation software solution project
- Refactored the software development process in order to deliver the product deadline, without any open customer issues and overtime – a first in the company's history (and with over 4 million licenses sold)
- Technologies used: C++, Python, TeamCity, subversion

**Software Developer**                           **TTL LLC.**                           **2013 – 2014**

- Feature development and maintenance on an automotive line-fit navigation software product
- Successfully designed and implemented multiple communication interfaces (including speech recognition interface with 3rd-party vendor)
- Technologies used: C++, Python, TeamCity, subversion

**Test Lead**                                   **Ericsson Communications**                           **2012 – 2013**

- Planned and scheduled functional tests, mentored junior test engineers on trainings and one-on-one basis
- Sped up the provocative test case creation process via automation, reducing development time for customer case test cases by 40–80%
- Technologies used: Bash, PowerShell, C, TNSDL, C++, Qt

**Software Engineer**                           **Ericsson Communications**                           **2011 – 2012**

- Manual and automated testing
- Developed and tested the implementation of multiple LTE features (VoLTE features through SIP protocol)

**Intern**                                   **Ericsson Communications**                           **2011 winter**

- Web-based tool development
- Technologies used: JavaScript, HTML, CSS, Bash, mySQL

---

## Analysis of the refactored resume

The refactored resume performed far better. This person heard back from companies who ignored their previous resume. They did well on multiple interviews and had several offers to choose from in a matter of weeks.

There's always room for improvement, though, and I would still suggest the following changes to make this CV even stronger:

- **More consistent formatting** as a whole. The spacing and bullet points have a strange layout.
- **Date formatting** could be cleaner. Using seasons is fine for e.g. internships ("summer 2011"), but instead of "2019 summer - 2020 spring", "Jun 2019 - Mar 2020" reads more clear.
- **Use active language.** Instead of "Creating backend microservices...", say "Created backend microservices...".
- **No need to repeat the header** on the second page - it just wastes space.
- **Education:** move it further down. With a degree almost 10 years ago, this is far less relevant than their work experience.
- **Consider using a better template or a resume template service** like Standard Resume for a more professional layout. This person went alone with the template, customizing it themselves. It still feels very DIY, and resume layout is not their strength.

---

## SRE Engineer with 20 Years Experience

The resume is from a systems admin working in a high performance computing environment, who holds a PhD. They have been happily employed for over 10 years at their current workplace. Their original resume was just adding details of the last 10 years' employment to the same resume they used when applying in 2010.

Following the refactor, this person saw a good amount of interest. They ended up getting and accepting an SRE offer - exactly the position they were after.

### **The original resume**

**SAM SMITH BSc, MSc**

**Address:** 163 Dourn Road, London SW6, UK  
**Phone:** +441 23 2456789  
**Email:** sam.smith\_@gmail.com

**PERSONAL PROFILE**

Extensive Linux Systems Administration experience, with 10+ years in a High Performance Computing environment.

Network and systems administration of multiple large compute clusters, including taking the systems from design and tender, through to physical installation, software stack, provisioning, management and maintenance. Monitoring, performance analysis, and working at scale are core parts of current role.

Deployment of physical and virtual servers and services, including high availability services.

Management and monitoring of private cloud infrastructure.

Excellent communication, interpersonal and technical writing skills, backed up with IT training certifications and extensive IT training experience.

Code development in multiple languages.

Have a proven interest in learning new technologies, languages and systems. Completed a 1 year full-time Taught Masters in HPC while working full-time in systems administration.

**KEY SKILLS****Operating Systems**

- Linux: RHEL, Centos, Ubuntu (strong)
- PaaS, IaaS, Cloud & Virtualisation (strong)
- Windows Server & Active Directory (prior experience)

**Training & Customer Support**

- Problem-solving and troubleshooting of the 'full stack': hardware, middleware, software
- Excellent interpersonal skills
- Training course design and delivery
- Technical writing

**Programming Languages**

- C, Perl, Bash (strong)
- Python, PHP (proficient)
- Go, Java, JavaScript (basic)

**Technologies**

- Core networking: TCP/IP, DNS, DHCP, HTTP, TLS, Firewalls
- LAMP stack
- Orchestration using Puppet, Ansible, Salt
- Monitoring via Prometheus, ELK, Nagios, Cacti
- High availability: HAProxy, Keepalived
- Virtualisation using OpenNebula, KVM, Ceph.
- Git and basic CI/CD

**CAREER SUMMARY**

**Senior Systems Administrator**      **TPC**, University College London, Bloomsbury, London, UK  
 Feb 2010 – Present

- Design, installation, management, maintenance and monitoring of multiple high performance Linux compute clusters, networking, SAN storage.
- Team lead since Nov 2012.
- Manage 300+ physical Linux servers across three data centres.
- Infrastructure as code (physical, virtual) using deployment and orchestration technologies such as PXE, Kickstart, Puppet, Ansible, Salt. Canary deployments for node upgrades.
- Systems monitoring using Prometheus, ELK, Nagios, Cacti and custom health check scripts.
- Security and user management including LDAP database provision and maintenance.
- Automated deployment of TLS certs using Let's Encrypt and HAProxy.
- Resource management and scheduling on compute clusters, including development of multiple custom add-ons to scheduler.
- Provision of research support to researchers from multiple disciplines, and to the community in general through online documentation and training.
- Liaising directly with research groups, advising on hardware and software requirements.

- Compilation of scientific codes and libraries, occasionally involving patching.
- Design, development and delivery of training courses.
- Teaching of programming and software skills.
- Liaising with vendors and service providers.
- Active participation in research projects in web development & CMS deployment.
- Collaborate on cross-functional projects within the larger IT unit.

**Co-Founder and Trainer** **researchlab**, 4 Rock Drive, London, UK  
Oct 2008 – Dec 2009

- Set up a new IT training business focused on designing and delivering bespoke, high-quality, high-impact training to organisations.
- Delivered training to individuals and teams from numerous Irish/multinational organisations, including blue-chip IT companies and government departments/agencies.

**Senior Training Consultant** **Tents Ltd**, St Johns, London, UK  
Jan 2004 – Sep 2008

- Provided in-house training and supervision to junior training staff.
- Researched both existing and new IT topics, developing course-ware and training materials.
- Delivered training to individuals and teams from numerous Irish/multinational organisations, including blue-chip IT companies and government departments/agencies.
- Courses included: MCSA, Linux Admin, Network+, A+, Java, Perl, PHP, JavaScript, Crystal Reports.

**Systems Administrator** **Futuristic Systems**, Graham House, 150 Upper Thomas's St, Liverpool, UK  
June 2000 – Sept 2002

- UNIX and NT systems administration, involving installation and maintenance of PCs, Sun workstations & servers, and switches.

**Systems Administrator** **The Internet Society**, Liverpool, UK  
Sept 1998 – May 2000

- Installation and maintenance of UNIX systems such as Linux PCs and Sun Solaris servers, on a voluntary basis in the student-run college Internet society.

## EDUCATION

**Postgraduate Degree:** **University College London** Oct 2011 – Sep 2012

**Masters Degree (Taught) in High Performance Computing, School of Mathematics**

Taught masters focusing on practical training in the use of large-scale and multi-core computers for scientific simulation and mathematical modelling. Course work included numerical analysis, parallel programming (MPI), stochastic modelling, Monte Carlo simulation.

**Postgraduate Degree:** **University College London** Oct 2001 – Dec 2003

**Masters Degree (Research) in Computer Science.**

Research project focused on WWW Connectivity Analysis, and in particular Web Spider technology. Project work involved Java, SQL, JDBC, Linux, HTML, and HTTP.

**Undergraduate Degree:** **University College London** Oct 1997 – Jun 2001

**Joint Honours Degree (1st class honours) in Computer Science and Mathematics.**

## IT CERTIFICATIONS

- |  |  |
|--|--|
| <ul style="list-style-type: none"> <li>▪ MCSA on Windows Server 2003</li> <li>▪ MCSA on Windows 2000 Server</li> <li>▪ Microsoft Certified Professional (MCP)</li> </ul> | <ul style="list-style-type: none"> <li>▪ ECDL Certified Training Professional (CTP)</li> <li>▪ CompTIA Network+</li> <li>▪ CompTIA Certified Technical Trainer (CTT+)</li> </ul> |
|--|--|

## Analysis of the resume

This resume feels dated just by glancing at the template. Unfortunately, subconscious biases can be a thing in tech, and some hiring managers might wonder if this is a person who would keep up with technology. Having interacted with the person, I can tell you they are - but their resume does not indicate this.

### The good

- **A very experienced candidate.** With a PhD and close to 20 years of professional experience, this candidate has a lot of things to bring to the table.
- **Lots of relevant technologies.** HAProxy, Prometheus, ELK and other technologies they've listed are considered close to cutting edge in 2020. This person clearly keeps up to date or works at a place that moves with the industry.

### Improvement areas

- **Too long of an intro** with too few specifics. Trim the introduction, and talk about results.
- **Irrelevant key skills** listed on the first page. "*Excellent interpersonal skills*" misses specifics, and "*Training course design*" is not a skill that is relevant for the positions this person is applying for.
- **Ten years in a job with no career advancement.** This person had been employed in the same place since 2010. Were they promoted? Or are they the most senior person already? The resume doesn't help answer this question. It should!
- **Irrelevant older positions** listed. Senior training consultant 2004-2008 doesn't add much to the resume, nor does the first few systems admin positions. The resume would read stronger if it omitted those: especially that this person is looking to apply for SRE positions.
- **Missing results of the work.** For an experienced candidate, it's expected they bring results to the table. The resume doesn't showcase these.
- **Cliches** in some of the descriptions like "*excellent interpersonal skills*", "*excellent communication*". Remove these, or replace them with specifics.

## The refactored resume

## Technologies and Languages

- Languages: C, Bash, Python, Perl (strong). Go, PHP, Java (proficient)
- Technologies: Linux, LAMP stack, TCP/IP, HAProxy, Keepalived, OpenNebula Cloud, Spectrum Scale
- Tools: Elastic Stack, Docker, Jenkins, GitLab CI, SaltStack, Ansible, Prometheus, Grafana
- Other: CI/CD, Parallel Programming, Cluster Management, High Availability, Load Balancers

## Work Experience

Senior Systems Administrator	University College London	2012 -Present
TPC	London, UK	
<ul style="list-style-type: none"><li>• Leads a team of 3 administrators, managing 300 physical machines and private cloud of 150 virtual machines across 2 data centres, providing virtualisation resources and 300TB of storage.</li><li>• Technical lead on management of 10 High Performance Computing Linux clusters, serving over 100 researchers from multiple faculties. Provides over 42 million CPU-hours per year.</li><li>• Identified legacy monitoring and alerting tools (Nagios, Cacti) as not being fit-for-purpose with modern systems and approaches: alert fatigue, not alerting on symptoms, lag time before alert fires. Led introduction of more modern tooling: Prometheus, Grafana, Elasticsearch (ELK).</li><li>• Identified frequent manual cluster node remediation as a source of toil. Improved overall cluster uptime from 90% to 95% by leading the creation of auto-remediation tooling for common issues. Wrote automated burn-in tests as additional checks before bringing the node back online.</li><li>• Reduced runtime of cluster node health-checks from 3 seconds to 0.3 seconds by introducing caches of system hardware configuration.</li><li>• Configured Linux I/O tuning parameters for parallel file system performance on HPC nodes.</li><li>• Created CI/CD pipeline for building core HPC cluster management packages, using Docker with Jenkins and Gitlab CI. This reduced time-to-deploy from hours to minutes.</li><li>• Wrote 70% of our cluster automation tooling in the past 5 years. Reviewed the remaining 30%.</li><li>• Contributed 45% of changes to our infrastructure-as-code configuration and tooling. Reviewed the remaining 55%.</li><li>• Wrote tooling to automate most common user queries: password resets, account creation, HPC project management lifecycle and cluster resource provisioning. Common queries can be closed in minutes rather than hours.</li><li>• Built tooling for canary deployments to HPC cluster nodes, which improved confidence in upgrade process and increased upgrade velocity from a couple of times a year to at least once a month.</li><li>• Identified manually-configured virtual machines as an operational problem, and leveraged our existing HPC-based infrastructure-as-code tooling to automatically configure hosts using SaltStack. Deployment times reduced from hours to minutes.</li><li>• Identified that cluster head-nodes were not included in the provisioning system. Simplified cluster upgrade procedures by writing additional configurations to include them. Reduced downtime for upgrades from hours to minutes.</li><li>• Introduced practice of running blameless post-mortems after incidents, as a learning practice and source of documentation.</li><li>• Reduced team overload by introducing practice of rolling triage role to manage incoming support tickets. Previously this was done by one team member.</li></ul>		

---

## Systems Administrator TPC

University College London  
London, UK

2010 -2012

- Built custom node provisioning system using core open source technologies PXE, DHCP, DNS, SaltStack.
- Introduced version control for all core infrastructure and internal tooling.
- Stood up 2 x 1000-core HPC clusters to provide 30 million CPU hours per year to over 100 researchers.
- Identified lack of monitoring tools for parallel file system (Spectrum Scale / GPFS). Wrote tools to feed into Nagios alerting system.

## Education

- |   |            |
|---|------------|
| • MSc. Mathematics, University College London, UK.                      | 2011 -2012 |
| • MSc. Computer Science, University College London, UK.                 | 2001-2003  |
| • BSc. Computer Science and Mathematics, University College London, UK. | 1997 -2001 |

## Projects and Community

- **MoxMoney**—add-onto HPC workload management software. Adds 'banking' facility to simplify allocation of CPU hours for administrators, and to improve reporting for end users. See the source on [GitHub](#).
- **Mox Tasks**—add-onto HPC workload management software. Adds ability to easily batch up multiple serial tasks in a task-farming workload. Improves compute node utilisation by at least 8x in common configurations. See the source on [GitHub](#).
- **IPAM-lite**—tool for identifying common issues with vanilla DHCP and DNS (BIND) data. Reports on: forward/reverse record mis-matches; identifying empty blocks of IP addresses; malformed DHCP entries; hostnames in DHCP which don't resolve; stale IP address allocations (combined with arpwatch data). See the source on [GitHub](#).
- Contributed patches to Open Source projects: Cacti, Nagios, Drupal modules.
- Conference speaker: 2013(Madrid), 2016(London), 2017(Liverpool).

## Teaching and Training

Masters in High Performance Computing, School of Mathematics, University College London

- Teach an annual 5-credit course: Tools for Scientific and Technical Computing.
- Course topics: Using Linux Command-Line; Version Control; Makefiles and build automation; Using Profilers and Debuggers; Bash Shell Scripting; Unit Testing.
- Taught 120 students since 2013.
- Supervised 5 30-credit dissertation projects since 2016.

Training courses of Research IT unit in University College London

- Organise training schedule for the Research IT unit.
- Manage multiple training events per year, with up to 15 attendees per course.
- Design, develop and teach courses including: Linux Command-Line, Bash Shell Scripting, Using the HPC Clusters, Debugging and Profiling in C, Using Version Control and Makefiles.

## Interests

- I am a keen amateur runner and have completed a number of marathons.
- I read as much as time allows, mainly science fiction.

## Analysis of the refactored resume

The refactored resume addressed all the above improvement areas:

- **An easier to read template** than before that also feels more modern.
- **Technologies and languages** jump out when starting to read the resume. It's clear that this person has experience with many of the latest distributed computing technologies.
- **Using active language** when describing their work experience.
- **The impact and results** come across far better in the work experience section. The language used is active, and there are lots of specifics.
- **Promotion is called out** in their current position. Instead of 10 years and no promotion, the story is more clear: promotion after 2 years, and they are now leading the HPC team, likely being the most senior HPC administrator.
- **Cutting out older positions** that don't add in helping with the job search. The resume is cleaner and more focused. And as a hiring manager, I wouldn't really care what this person worked on before they MSc, as in the decade since then, they've shown solid results.
- **Projects, teaching and training add to the resume.** The projects linked are solid, with the GitHub repos linked in the resume. Both conference talks and teaching at the university close off this strong resume nicely.

As an interesting point, even the refactored resume is two pages. This works for two reasons. First, this person is applying in the UK, for more mature organizations, where the one-page resume is not that common. Second, they have enough meaningful experience to fill the pages up with Ten years of work experience does warrant a full page.

In the end, the refactor most likely helped this person stand out from the crowd, telling a much stronger story than the original resume did.

## Other Real Resume Examples

[Standard Resume](#) shares real software engineering and engineering manager resumes that you can browse for inspiration on the content, and templates that you can use with the service.

- [Software Engineer](#) (Jeff Leu, eBay) and [other software engineer resumes](#)
- [Web Developer with bootcamp background](#) (Alexa Vita)
- [Web Developer](#) (Cameron Wardzala) and [other web developer resumes](#)
- [iOS Developer](#) (Zac West, Dropbox) and [other iOS developer resumes](#)
- [Frontend Engineer](#) (Heather Vandervecht) and [other frontend engineer resumes](#)
- [Data Scientist](#) (Philip Sanoudes, Square) and [other data scientist resumes](#)
- [Engineering Manager](#) (Mike Douglas) and [other engineering manager resumes](#)

# Chapter 14: Advice for Hiring Managers on Running a Good Screening Process

This chapter is for all fellow current or future hiring managers. I realize most people reading the book are not hiring managers, but one day you might become a lead or a manager, and have a say, or design a hiring process. By that time, you might have forgotten how difficult it was to put your foot in the door, and how much depended on having a really strong resume.

**As a hiring manager, invest time and energy to create a better and more fair screening process.** I've seen many hiring managers assume they have little to do with how resumes are screened. Many of them were often unaware of how recruiters and sourcers filtered the incoming resumes, and how they ended up rejecting very promising candidates before even a recruiter call—because they had no support from the hiring manager. Don't be this person. Here are a few things you can—and should—do for a better recruitment process.

## Know that your screening process is broken

People who read this book and apply the advice to their resume will likely fare better in their job search process, and progress through more resume screens. You might end up hiring one of these people, who turn out to be a great hire. And you would have missed out on them if they had a poor resume that you or your recruiter would have rejected.

In the book [\*Smart and Gets Things Done\*](#), Joel Spolsky says the same:

*"The standard job application, a cover letter and a resume, is a phenomenally weak way to introduce a candidate. It gives you only the faintest clue as to the quality of an applicant. Sometimes, though a resume gives pretty strong negative clues as to the quality of an applicant (...) Other than that, though, it can be extremely hard to tell much about a candidate from a resume."*

**As long as you have resume screening in place, you will always reject good people with poor resumes.** This is a fact that you won't be able to change—unless you relax, or drop the resume screening requirement.

However, the resume filter has a good reason to be in place: it reduces the number of applicants, filtering out people who are likely not a fit. As you are doing resume screening, make conscious choices on whether the resume is in the "Yes", "Maybe" or "No" piles. Following this approach, you can avoid early rejection. As [\*Johanna Rothman\*](#) explains in the book [\*Hiring Geeks That Fit\*](#):

*"Because I deal primarily with hiring technical people who may not know how to best present themselves in a resume, I read resumes with an eye to giving each person a benefit of the doubt—which leads to the Maybe pile. I re-read the Maybe resumes after I've phone-screened the Yes candidates but still have not filled the opening, to see if I've missed a diamond in the rough."*

This observation is what is fueling several startups to switch from a resume-based first screen to an assessment-based first screen. Triplebyte, Hackerrank and [\*Woven\*](#) are good examples of each, and you should consider if you want to invest in a similar approach.

**Solutions to reduce false negative screens in your hiring pipeline** like [\*Woven\*](#) offer to take all your rejected resume screens and assess them with an hour-long, hands-on exercise. They find people

you would have rejected, and who have applied for your position. Companies like Woven are proof that resume screening is broken, and they offer a pragmatic, hybrid approach. They find the “hidden gems” who are not represented fairly by their resumes, and claim to find [a third of hires](#) made this way—who otherwise would have been rejected. You progress with people with strong resumes, while Woven looks much closer at people who would have been rejected, spending more time and effort, but resulting in rejecting fewer people who would thrive in your team.

**Engineer job search platforms** like [Triplebyte](#) attract developers with the promise that if they pass the assessment done by the platform, they will go directly to onsites for various companies. Triplebyte focuses little on the resume, and gathers signals from assessments tailored for specific domains and technologies. Partnering with Triplebyte or a similar company means you could get access to talent who either would have not applied to you, or you might have rejected them at the resume screen.

**Coding challenge platforms** like [Hackerrank](#) offer coding challenges that you can configure as your primary screening method—often replacing resume screening. When using Hackerrank, I suggest taking care in setting up exercises that mimic the type of skills you’ll need to use day-to-day. [Don’t go overboard with algorithms](#) that people won’t ever need to use. Consider investing engineering time, both to build the challenges and to evaluate them. If you take one off the shelf, and have your recruitment team evaluate it based on the automated score that Hackerrank gives, you might miss out on people who wrote good code, but missed one small edge case.

**Even when you don’t have additional budget to invest**, you still should aim to minimize rejecting otherwise great applicants purely based on their resume. A few things that you can implement to help with this are the following.

- **Involve at least one other person in screening resumes**, and screen the same set of resumes independently. What is the overlap between your and the other person’s “Yes” pile? This exercise can be useful both to calibrate expectations, as well as to reduce bias.
- **Budget dedicated time for hiring**—either for you or for someone else. When you don’t have dedicated time for hiring activities, you’ll be rushed and have poor results. Have dedicated time from resume reviews, to building the interview loop, all the way to preparing for interviews. If you don’t have this time, delegate the process to someone who does have it.
- **Adjust the time allocated for hiring relative to the number and importance of open roles.** You’ll want to spend far more time with hiring when you are about to double the team than when hiring one junior developer. Be mindful of creating the attention the hiring process should have—including screening of resumes.
- **Be aware of your cultural biases.** You will most likely have a [self-selection bias](#), subconsciously favoring people who are similar to yourself. Acknowledge this bias, and aim to select profiles that don’t resemble your background and experience.

Consider implementing the advice in the rest of the chapter for an even better recruitment process, starting with partnering with your recruitment function.

## Treat recruitment as a partner function

**Build a strong relationship** with recruiters, sourcers and inbound sourcers and treat them as true partners from when you start hiring. Sit with them, and understand how they move candidates between stages. Explain what types of traits you are looking for in your team, and what makes peoples successful. Have regular check-ins with this group, where you both share details.

If you don't reach out proactively, your HR or recruitment function might assume you're not interested in their work. I've often seen hiring managers assuming they are not welcome to help. Whenever I see hiring managers and HR/recruitment work in silos, it's a huge missed opportunity for making the recruitment process better.

Be curious: talk about sourcing strategies, key signals your recruitment team looks for, as well on the diversity of traits people on your team have. The more your recruitment team and you have mutual understanding of how you each work, the better your results will be.

**Empower recruiters and sourcers to** experiment with bringing promising people to technical interviews. Over time, recruiters and sourcers learn the type of people who make it to an onsite and get an offer, and ones that don't. Recruiters try to not waste engineering resources for candidates who don't meet all requirements set by the hiring manager. Don't be the hiring manager who does not negotiate on having 5 years' Java experience and 2 years' JavaScript. Come up with sensible guidelines, but also encourage your recruitment team to take the initiative, and champion people they feel stand out for one reason or another.

**Make yourself available to review resumes.** Some of the best hiring managers I know get their hands dirty and review resumes together with their recruitment team. Hiring managers can often read more into resumes that aren't written that well, but for example, have good code or challenging projects linked. Try to be this hiring manager. Even if you don't do it full-time, take the time to look through some of the rejected resumes, and see if there are people worth investing in moving to the next round.

### **From the inside out: great hires who did not meet the standard hiring criteria**

Hiring managers who don't go with strict hiring guidelines, and who empower their recruiters to think outside the box, almost always end up hiring hidden gems. Here are a few stories.

James Stanier, SVP Engineering at Brandwatch shares a great hire he's made: "*I remember one person I once employed had a year's career gap because they were a semi-professional bass player and went touring with a band. Initially that got looked down upon, but that's actually really freaking awesome! He was a great engineer.*"

Blake Stockman, who has been a tech recruiter at Google, Facebook, Uber and other tech companies, shares on the most unlikely hire he's made: "*There was one individual who reached out to everyone on the recruiting team, for weeks. They kept sending messages asking to talk about an internship. They persisted. Finally, I said: okay, okay, fine. Like, let's have a conversation. I hopped on a call with them. It turned out to be a really enthusiastic person who was really hungry for an opportunity. They were preparing for this for some months already and were really keen to see how they would measure up.*

*We were just kickstarting our internship program. I decided to take a chance on this person, and put them through the interview process. They just blew everyone out of the water. They ended up coming on as an intern, then converting full time, and are still with the company today.*

*The crazy thing is, almost all recruiters would have probably rejected them, if they had come through the resume pipeline. Their background did not stand out—at all. But because they were enthusiastic, eager, humble, willing to learn and persistent, they were able to get that opportunity. There are definitely more people like this, who shine when given the right opportunity. It has been very rewarding to see diamonds in the rough like them who are successful and grow—just because someone took a chance on them.”*

I also have several similar stories. One of the best engineering managers I've worked with was self-taught with no computer science degree. I was on the hiring committee for an engineer who was doing pizza deliveries two years before for a whole year—and the committee was trending towards not hiring him as this was “quite unusual”. They were a top hire that we made. I hired an Android engineer with only iOS experience, who was eager to learn. They aced the job.

And my story is similar: I was hired at Skyscanner to do iOS, without having built iOS apps full-time, though I had extensive Windows Phone experience. I picked up iOS in no time, on the job.

## Write a good job description

**Be involved** in writing and iterating on the job description. Some hiring managers end up copy-pasting existing job descriptions and calling it a day. This will be a poor experience for candidates, recruiters and interviewers.

**Challenge yourself on what is really needed for the job.** Before writing the job description, sit with your recruiters and hash out what you are *really* looking for. What would this person do in their first month, and their first year? What skills and experience are absolutely non-negotiable, and what can they pick up on the job?

For example, if you're looking to hire for a tech lead on the ads team where the team uses Node.js, having led a small team in some capacity is probably a must-have, but having worked in the ads industry is probably not. And the language or framework might be negotiable, when talking about someone who is a fast learner, has backend experience and is motivated in picking up Node.js. By removing this constraint, you'll also have a larger pool of qualified candidates to work with. This is the reason most of the big tech companies don't specify a specific language: hiring someone who has done similar work on a different stack can work just as well as hiring someone with deep domain expertise.

**Avoid bias and gendered phrasing in the job advert.** If you have job descriptions with masculine wording, then you'll have fewer female applicants, many of them being turned away just by the language. Aim to write inclusive and welcoming job adverts that use de-biased language. Accenture in the UK changed their job descriptions to be more inclusive, and saw an increase of female applicants from 34% to 50%, according to the [HR Magazine article on de-biasing language in job adverts](#).

*“There are two easy key ways to take the gender bias out of job ads. One, purge the gendered language. Two, limit the number of mandatory qualifications to apply for the job,” says Iris Bohnet, author of the book [What Works: Gender Equality by Design](#) in the Harvard Business Review article [Simple ways to take gender bias out of your jobs](#). Following this advice, be mindful of how many things are truly a “must have” in your job description, versus mentioning things that the people will learn in the role.*

You can use tools like the [Gender decoder for job ads](#) by Kat Matfield and get inspired by well-written advertisements like the [Engineering manager opening at Honeycomb.io](#).

## Have a feedback loop

**Monitor your hiring pipeline—and look behind the numbers.** What percentage of people are progressing through the various stages? How is this number changing over time? Once you have the numbers, don't fully rely on just these; keep your ears on the ground to see how hiring is progressing on a case-by-case basis as well.

**Monitor how diverse your recruitment and your hiring pipeline is.** Diverse teams have proven to deliver better results. As a hiring manager, your goal should be to hire in a diverse way. But are you getting a variety of people to apply? Do they make it through the resume screen? If you don't have data on this, you won't be able to act on it. So get involved in getting this data and pairing with your recruitment team.

**Have retrospectives on how things are going.** Assuming you are regularly talking with recruiters and sourcers, reflect on what each group is seeing. How is the response rate from sourcers? How have referrals been faring? What types of inbound applications are coming your way? What candidates are recruiters most excited about? How have the interviews gone, and is there any feedback that stands out? The more, and the more openly, you talk, the more you'll spot problem areas that you can decide to address.

Short, informal retrospectives after successful and failed onsites are also a good idea. These are both great learning opportunities. When you hired, ask yourself how you could expand your funnel to have more people with similar characteristics apply? When a person doesn't get an offer, reflect on whether there is anything that could have predicted this earlier. Did this person not get an offer due to not meeting an unwritten requirement? Should you make that part of the job description? Or was it a really close call, and should you encourage looking for more of this, and similar profiles?

**Have a clear policy on providing feedback at different stages of the process.** What happens when someone is rejected at the resume stage? What about when it's after the technical phone screen? Confirm what the process for feedback is with your recruitment team. You'll want a policy that is sensible, and takes any legal and company constraints into account. While it would be amazing to give honest feedback on every single resume, as hiring managers, we know that this is not feasible for many reasons—legal and the resources needed being some of them. Still, you'll want to ensure that candidates have a consistent experience, and you should also be clear when you encourage certain people to re-apply after a given time.

### ***From the inside out: the story of BusyManager and MakeTimeManager***

I observed two very different hiring managers in how they worked with in-house recruiters, and what their results were, over the course of a year.

BusyManager had a lot of things on their plate, and rarely had time for recruitment. They had a lot of fires to put out, shipping features, aligning with the business, and securing headcount. And they got a nice headcount grant, allowing their team to grow significantly. They did what most managers would do at this point: tasked recruitment to fill the headcount and let them run with it. They had more important things to do, and barely had time to put a short job description together. When they

finally got around to checking in, things were not moving. They made it clear to recruitment that they needed results, and fast. However, the recruitment team just couldn't get the right people on board: BusyManager rejected almost all candidates who were brought onsite. BusyManager changed recruiters a few times to get better results—but these results just did not come.

MakeTimeManager had a very small headcount to start with. Still, they wanted to make sure they hire right. The first thing they did was set up a weekly meeting with the recruitment team. They talked about expectations, opportunities and progress on this meeting. MakeTimeManager made it clear to the recruitment team that hiring comes first for them, and they are always free to interrupt.

They got lots of interruptions for resume screening, closing candidates and so on; but MakeTimeManager stood by their word. They were involved in all parts of the process, all the way to ensuring candidates had a great onsite experience. They advocated for all their team members to put recruitment first, always make time for interviews, prepare for them, and put the candidate experience above other tasks.

MakeTimeManager hired quickly, and hired well, surpassing all expectations. They were given more headcount after filling their original one. BusyManager failed to fill their headcount by a large gap. And to complete the picture: both managers worked with the exact same set of recruiters.

So how was it that only MakeTimeManager was getting results? It was because of the two, MakeTimeManager was the one who recognized that treating recruitment as a partner is key in growing any team, and doing it well. While recruiters loved working with MakeTimeManager, they found it hard to understand what BusyManager wanted to hire for, and the results reflected on this. When you're hiring, take the example from MakeTimeManager and catch yourself when you are starting to act like BusyManager.

## Learn how others do it

This book is not about how to hire developers well, but several books cover this in detail. I recommend the following reading:

- [\*Smart and Gets Things Done\*](#) by Joel Spolsky is a concise, opinionated and good guide on how to hire good developers. Though it was written in 2007, there are not many things that are dated about the book's contents.
- [\*Hiring Geeks that Fit\*](#) by Joanna Rothman. A must-read if you're hiring for the first time, or if you don't yet have an organized hiring strategy. The book guides you through defining a hiring strategy, analyzing the job and planning and executing all parts of the process. It ends with reference checks, making an offer and creating a great first day for the newly hired person.

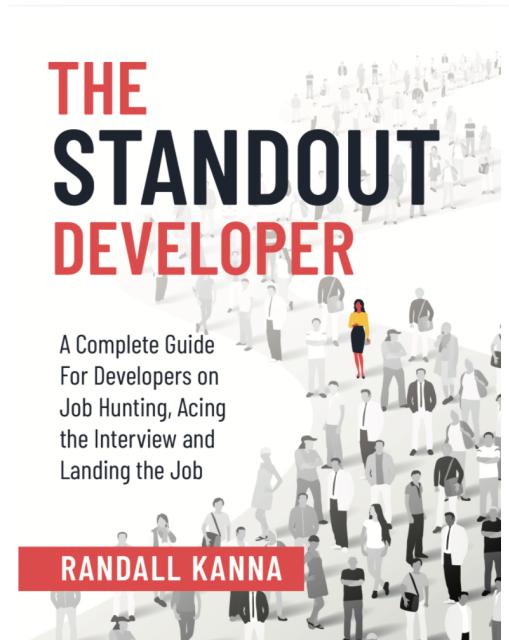
## Further Reading

This book covered crafting a resume. While the resume screen is often a large “drop off” point for most applications, the hiring process is challenging in the following stages as well. While this book deliberately does not touch on what comes after, below are books that you might find helpful in the other parts of your job hunt. These are books that I have read, and recommend.

- [\*\*\*The Standout Developer\*\*\*](#) by Randall Kanna

Where this book ends, Randall’s book carries on. Randall has broken into software development with no CS background, following a bootcamp. She’s since worked at the likes of Pandora and Eventbrite, and shares her journey and advice on how to follow her example.

Randall also shares a [free job hunting guide](#), with several resources that could help on your job hunt.



- [\*\*\*Grokking Algorithms\*\*\*](#) by Aditya Y. Bhargava

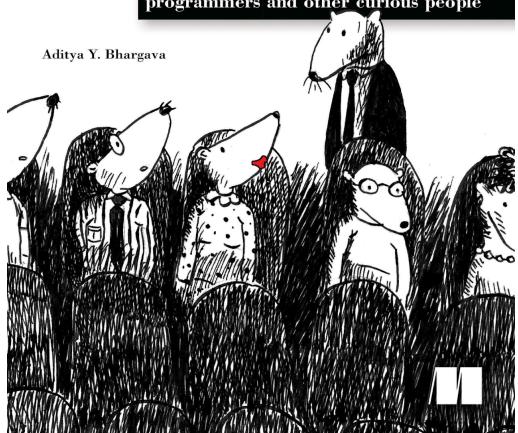
This book is hands down the best guide on algorithms from beginners to experienced engineers. A very approachable and visual guide that covers all that most people need to know on this topic.

I am convinced that you don't need to know more about algorithms than this book covers. I wrote about my thoughts on algorithmic interviews in the article [Data Structures and Algorithms I Used Working at Tech Companies](#).

**grokking  
algorithms**

An illustrated guide for  
programmers and other curious people

Aditya Y. Bhargava



- **[Cracking the Coding Interview](#)** by Gayle Laakmann McDowell

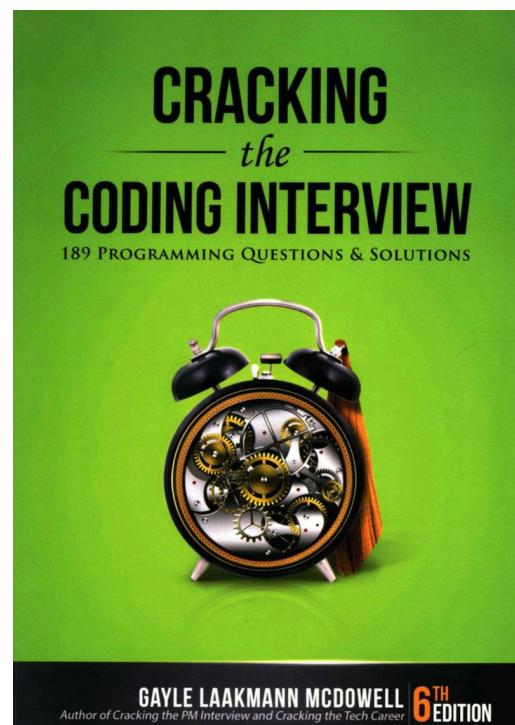
Many large tech companies and startups ask coding challenges where data structures and algorithms knowledge are often needed. Cracking the Coding Interview is by far the most popular book to prepare for this.

I've gone through the book when preparing for interviews at Facebook and Uber and recommend it to prepare for the "Silicon Valley"-type interviews. Work your way through it: don't skip to solutions.

While there are many online platforms that let you practice problems. This book is not only my recommendation: most recruiters I talk with, who hire for larger tech companies, suggest this resource for candidates.

- **[System Design Interview - An Insider's Guide](#)** by Alex Xu

When preparing for the systems design interview—asked at tech companies, usually for senior or above roles—there are few good resources to prepare. Book-wise, I've found this book from Alex Xu, who worked at Twitter, Zynga and Oracle, to be a solid preparation resource. The book is also available as [an online course](#).



**[I am writing a new book](#)** on growing as a software engineer, from junior to staff/principal positions, expected to be released early 2021. [Sign up here](#) to get notified when it is out.

---

## Conclusion

This is it: we've finished with all the advice I had to share on crafting a tech resume that represents you fairly. We went from looking "behind the scenes" at the hiring pipeline, then resume basics, structure, mistakes to avoid, and exercises to polish your resume. We analyzed characteristics of a good resume template and looked at "before" and "after" examples for resumes that did better after they were refactored.

The resume screen is the very first step in an otherwise long hiring process. I hope this book and its advice will serve you well in clearing this first step at more places than you did before.

If you have success stories to share after updating your resume, or have other feedback, I'd love to hear from you at [hello@thetechresume.com](mailto:hello@thetechresume.com).

All the best on your journey!