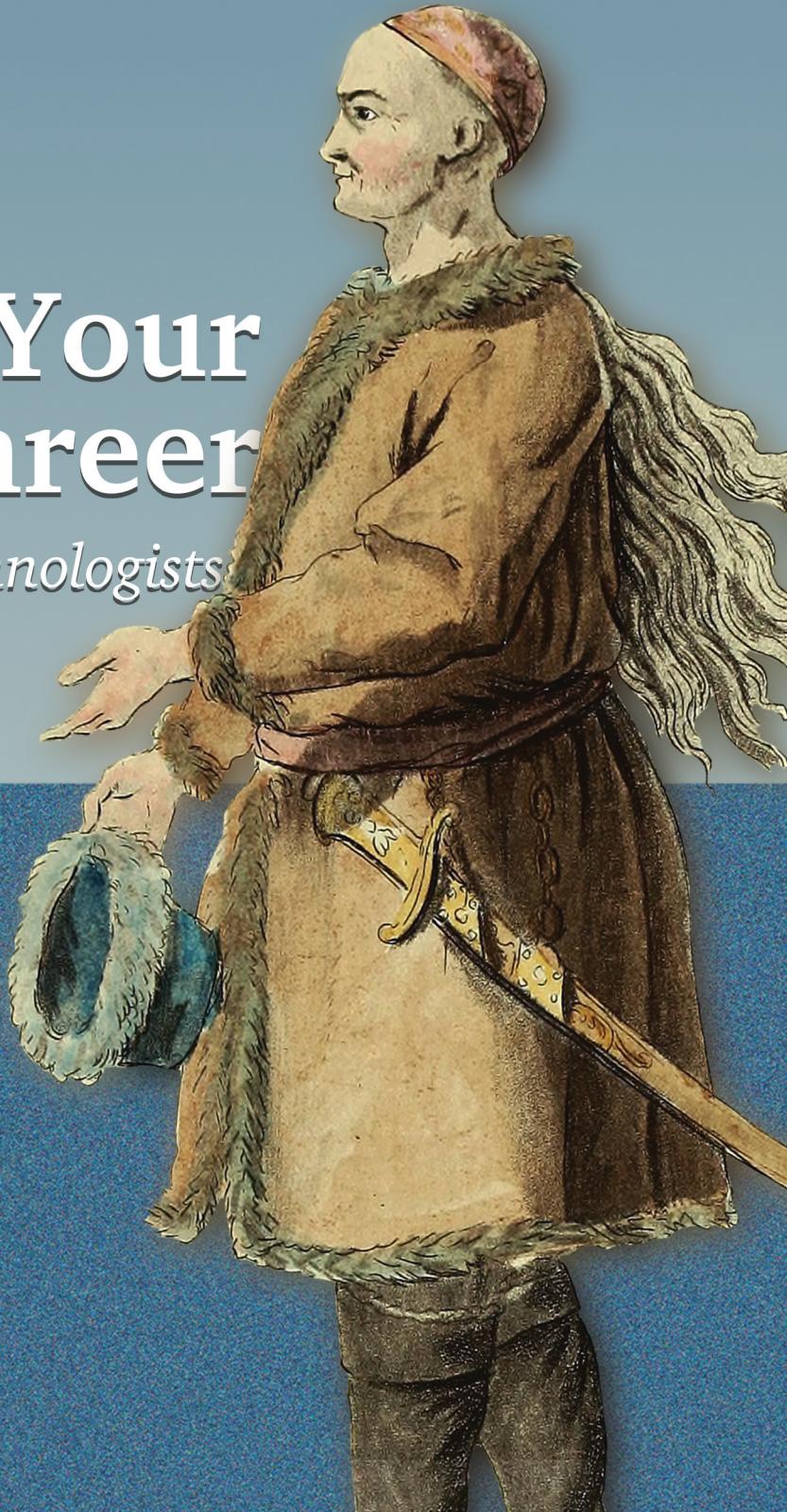


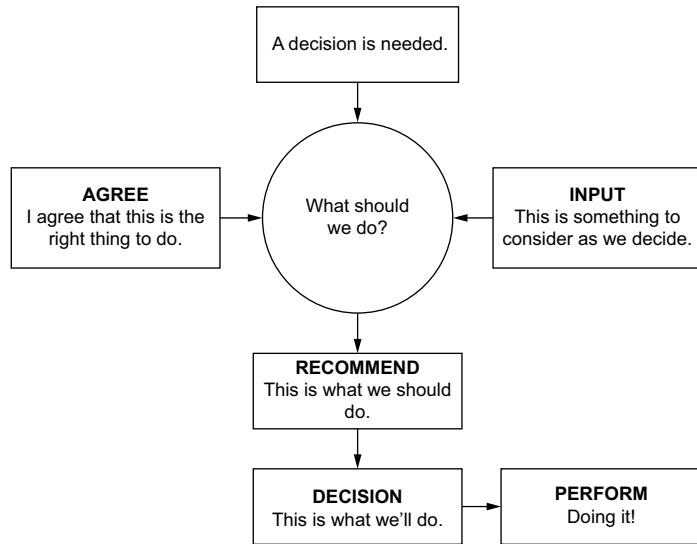
Don Jones

Own Your Tech Career

Soft skills for technologists



MANNING



Being a better decision maker and understanding how businesses make decisions is critical to taking your career in whatever direction you want. RAPID is a popular business decision-making framework, and at a high level this chart illustrates the various roles that participate in the framework.

Own Your Tech Career

Own Your Tech Career

SOFT SKILLS FOR TECHNOLOGISTS

DON JONES



MANNING
SHELTER ISLAND

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

©2021 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

⊗ Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.

 Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Development editor: Frances Lefkowitz
Technical development editor: Chris Villanueva
Review editor: Aleksandar Dragosavljević
Production editor: Keri Hales
Copy editor: Keir Simpson
Proofreader: Melody Dolab
Typesetter: Dennis Dalinnik
Cover designer: Marija Tudor

ISBN: 9781617299070
Printed in the United States of America

brief contents

- Introduction 1
- 1 ■ Own your career 3
- 2 ■ Build and maintain your brand 12
- 3 ■ Network 21
- 4 ■ Be part of a technology community 30
- 5 ■ Keep your tech skills fresh and relevant 37
- 6 ■ Show up as a professional 49
- 7 ■ Manage your time 58
- 8 ■ Handle remote work 66
- 9 ■ Be a team player 75
- 10 ■ Be a team leader 84
- 11 ■ Solve problems 97
- 12 ■ Conquer written communications 104
- 13 ■ Conquer verbal communications 120
- 14 ■ Resolve conflicts 130
- 15 ■ Be a data-driven, critical thinker 138
- 16 ■ Understand how businesses work 147

- 17 ■ Be a better decision-maker 163
- 18 ■ Help others 178
- 19 ■ Be prepared for anything 188
- 20 ■ Business math and terminology for technologists 195
- 21 ■ Tools for the modern job hunt 209

contents

preface xv

acknowledgments xvii

about this book xviii

about the author xx

about the cover illustration xxi

Introduction 1

1 *Own your career 3*

- 1.1 Job, career, success, and self 3
- 1.2 Start at the beginning: With yourself 4
- 1.3 What does success look like for you? 7
- 1.4 Creating a career plan for right now 9
- 1.5 Action items 11

2 *Build and maintain your brand 12*

- 2.1 Brand building: Know your audience 12
- 2.2 Social media and your brand 15
- 2.3 Your brand has a wide reach 17
- 2.4 Professionalism and your brand 18
- 2.5 How to sabotage your brand 19

- 2.6 Further reading 19
- 2.7 Action items 20

3 Network 21

- 3.1 Why networking? 21
- 3.2 The problem with digital communications 22
- 3.3 Ideas for in-person networking 23
- 3.4 Ideas for online networking 24
- 3.5 Etiquette for networking 25
 - In person* 25 ▪ *On LinkedIn* 27
- 3.6 Becoming a confident networker 28
- 3.7 Action items 29

4 Be part of a technology community 30

- 4.1 The value of community to your career 30
- 4.2 Yes, you're worthy of contributing 32
- 4.3 Ways to contribute and participate 33
- 4.4 Etiquette for participating 34
 - On Q&A websites* 35 ▪ *In open source projects* 35
- 4.5 Action items 36

5 Keep your tech skills fresh and relevant 37

- 5.1 Fresh vs. relevant 37
- 5.2 Deciding what's relevant 40
 - Proficient, not expert* 40 ▪ *Where to focus* 43
- 5.3 Building strong learning muscles 44
- 5.4 Learning media 45
- 5.5 Assessing your relevancy 46
- 5.6 Tips for lifelong daily learning 46
- 5.7 Further reading 47
- 5.8 Action items 47

6 Show up as a professional 49

- 6.1 Be your word 49
 - Never promise what you cannot deliver* 49 ▪ *Always deliver what you promise* 50 ▪ *Be easy to work with* 50

- 6.2 Be detailed and precise 51
- 6.3 Cut your losses when the time is right 53
- 6.4 Let Blue Sky mode happen 54
- 6.5 Draw a yellow line 55
- 6.6 Action items 57

7 *Manage your time 58*

- 7.1 Discipline, procrastination, and laziness 58
- 7.2 Time management 59
 - Time inventory: The TimeFlip technique 59 □ Time rationing: The Pomodoro technique 61 □ Time catalog: Knowing your capabilities 63*
- 7.3 Multitasking 63
- 7.4 Action items 65

8 *Handle remote work 66*

- 8.1 The challenges of being remote 66
- 8.2 Creating a space 67
- 8.3 Creating a space when you have no space 68
- 8.4 Working with family 70
- 8.5 Adopting a routine 71
- 8.6 Explicitly defining a culture 72
- 8.7 Networking like you're in the office 73
- 8.8 Remote work: Permanent or temporary? 74
- 8.9 Action items 74

9 *Be a team player 75*

- 9.1 The ups and downs of teams 75
- 9.2 A checklist for being a better team player 76
- 9.3 Dealing with less-effective teams and teammates 79
- 9.4 Contributing to an inclusive workplace 80
 - Help 80 □ Offer respect and support 81*
- 9.5 Further reading 82
- 9.6 Action items 83

10 Be a team leader 84

- 10.1 The decision to lead 84
- 10.2 Leadership vs. management 85
- 10.3 The leader's path 86
- 10.4 Getting into their context 88
- 10.5 Leading positively 91
- 10.6 Mistakes leaders make 92
- 10.7 Leadership beyond leading 93
- 10.8 Before moving into leadership 94
 - Don't get promoted to your level of incompetence* 94 ▪ *Learn leadership* 95 ▪ *Measure your own success* 95
- 10.9 Further reading 96
- 10.10 Action items 96

11 Solve problems 97

- 11.1 Problem-solving vs. troubleshooting 97
- 11.2 Clearly state the problem 97
- 11.3 Identify your levers 99
- 11.4 Negotiating solutions 101
- 11.5 Action items 102

12 Conquer written communications 104

- 12.1 Communicating is telling a story 104
 - The rules of storytelling* 105 ▪ *Applying storytelling to business communication* 106 ▪ *What about mundane, everyday communication?* 107
- 12.2 Facing our fear of communicating 108
 - Analyze the causes of your fear* 109 ▪ *Address the causes of your fear* 110 ▪ *Conquer fear in written communications* 111
- 12.3 Applying structure to your storytelling 113
- 12.4 Practice, practice, practice 116
- 12.5 Common written defeaters 117
 - Avoid passive voice* 117 ▪ *Prune that flowery garden* 118
- 12.6 Action items 118

13 Conquer verbal communications 120

- 13.1 Stepping up to verbal communications 120
- 13.2 Conquering your fear of speaking 121
 - Fear of not having all the answers 121 ▪ Fear of being judged 122*
- 13.3 Common verbal defeaters 123
- 13.4 Finding the right amount of assertive 126
- 13.5 Persuasion and the art of listening 127
- 13.6 Action items 128

14 Resolve conflicts 130

- 14.1 Conflict can be healthy and even deliberate 131
- 14.2 Seeking context 131
- 14.3 Returning to first principles 134
- 14.4 Relying on data 135
- 14.5 Using decision-making frameworks 136
- 14.6 A win doesn't matter as much as the outcome 137
- 14.7 Action items 137

15 Be a data-driven, critical thinker 138

- 15.1 In business, never "believe" 138
- 15.2 Be a data-driven, critical thinker 139
- 15.3 Be data-driven 144
- 15.4 Beware the data 145
- 15.5 Further reading 146
- 15.6 Action items 146

16 Understand how businesses work 147

- 16.1 Businesses are people too 147
 - Businesses and their relationships 148 ▪ Customers and employees 149 ▪ One-sided relationships 149 ▪ Dealing with changes in the relationship 152*
- 16.2 How businesses really make money 153
- 16.3 What does your business sell? 156
 - Example 1: Terri's International Bulbs 156 ▪ Example 2: Martin's Theme Parks 157 ▪ Example 3: Pat's Fruity Clothing 158 ▪ Know the details of the business 159*

- 16.4 Understanding risk and reward 159
- 16.5 Further reading 162
- 16.6 Action items 162

17 *Be a better decision-maker 163*

- 17.1 Deciding who decides: Decision-making frameworks 163
- 17.2 Deciding what to do: OKRs, rocks, and pebbles 166
 - Rocks and pebbles 166 ▪ OKRs 167 ▪ Priorities, priorities 168*
- 17.3 Deciding what to drop: Opportunity cost 169
- 17.4 Deciding what's enough: Good, better, best 170
- 17.5 Deciding what to believe: Being data-driven 173
- 17.6 Deciding together: How to negotiate 175
- 17.7 Further reading 177
- 17.8 Action items 177

18 *Help others 178*

- 18.1 Why help? 178
- 18.2 Yes, you can 179
 - The toxic relationships that keep us from teaching 179*
 - You are definitely worthy of teaching 179*
- 18.3 How humans learn 180
- 18.4 The value of repetition 181
- 18.5 Getting in and doing it 182
- 18.6 Why analogies work . . . and how they can fail 183
- 18.7 Do it like Socrates 183
- 18.8 The importance of sequencing 185
- 18.9 Rest time is crucial 186
- 18.10 Further reading 187
- 18.11 Action items 187

19 *Be prepared for anything 188*

- 19.1 What can happen? 188
- 19.2 Basic preparedness goals 189
- 19.3 Cash on hand and credit 189
- 19.4 Social safety nets 190

- 19.5 Insurance 191
- 19.6 Prestaged job hunt tools 192
- 19.7 Action items 193

20 *Business math and terminology for technologists 195*

- 20.1 How much do you cost? 195
- 20.2 Reading a P&L statement 197
 - Revenue 197 ▪ Expenses 199*
- 20.3 Averages 200
- 20.4 OpEx and CapEx 201
 - Understanding the two types of expenses 201 ▪ Driving business decisions 202*
- 20.5 Business architecture 203
- 20.6 Further reading 207
- 20.7 Action items 207

21 *Tools for the modern job hunt 209*

- 21.1 Job hunt tasks to do now 209
- 21.2 Review your brand 210
 - Your professional brand and the job hunt 210 ▪ Reviewing your public footprint 211 ▪ How would others describe your brand? 212*
- 21.3 Update your résumé 213
 - Résumé rules 213 ▪ Starting your résumé 214 ▪ Every résumé is unique 215 ▪ Analyze the job posting 216 ▪ Writing your résumé 220 ▪ Formatting your résumé 224 ▪ Should you hire a résumé writer? 226*
- 21.4 Nailing the interview 226
- 21.5 Understanding compensation packages 227
 - Compensation package elements 228 ▪ Negotiating your compensation package 230*
- 21.6 Further reading 231
- 21.7 Action items 231

index 233

preface

This book came from the confluence of several seemingly disconnected events. I was about ready to prepare a fifth edition of *Be the Master*, an independently published book on choosing your own success and growing that to help others. I was also looking to revise *Let's Talk Business*, another independently published title focused on making business basics more accessible to technologists and other individual contributors. Finally, Manning reached out and said that they wanted to create a new “soft skills” book, one designed to help technologists with the non-technical aspects of their careers. The timing seemed perfect, and so this book—through several iterations and many revisions—was born.

Soft skills, as a phrase, totally downplays the importance of things like communicating, understanding why businesses do what they do, working on a team, developing leadership skills, and so on. These skills may not be “hard” technical skills, but it turns out *most companies worry more about getting the right soft skills* than they do about making sure their people know the latest version of C#, or Windows Server, or Tableau, or something else. Technical skills can be developed through training and experience, and in many ways—like certifications—can be measured, to a degree. But “soft” skills are harder to wrap an organization’s collective mind around, harder to measure, and harder to develop. Yet arguably, our ability to work with other human beings is the *most* critical skill for any kind of success.

This book isn’t designed to be the last “soft skills” learning you’ll ever have. It’s designed to be the first, or one of the first: a way to frame up the skills you’ll need to focus on, grow, and refine throughout your entire career. This book comes largely

from my experience and the experiences of close friends and colleagues, as those experiences were the best way I could think of to ensure the book was real-world and relevant.

Whether you're brand-new in your tech career or a decades-long veteran, I think you'll find that this book offers valuable perspectives, new things to think about, and new topics to include in your career management repertoire. Most important, this book tries to stress the importance of your career being *your* career: *you* define what "success" looks like, *you* decide how to make that success happen for you, and *you* benefit from that success in the end. I've tried to frame everything in a way that puts *you* in the driver's seat of your career, offering advice and observations but *not* directives.

I hope you enjoy, and I hope you find much success along the way!

acknowledgments

I want to thank my friends and family—especially Chris and Donavan—for putting up with the gnashing of teeth and furious typing as I made my way through the principal writing and the editing passes.

I also want to thank the early reviewers of this book and its outline, many of whom provided invaluable suggestions and insights that they'll see reflected in this final result: Adriaan Beiertz, Bill Bailey, Bobby Lin, Cameron Presley, Christopher Villanueva, Dave Corun, Irfan Ullah, Ed Lo, Fernando Corrales, Joe Ivans, Lee M. Cottrell, Marc-Anthony Taylor, Markus Braasch, Neil Croll, George Onofrei, Sergio Govoni, Vasile Boris, and Warren Myers.

And finally, I want to thank all of the readers of my prior efforts, including *Be the Master* and *Let's Talk Business*, who provided so much feedback, encouragement, and constructive criticism.

about this book

This book is for anyone pursuing or living a career in technology: developers, data engineers, network architects, systems operators, security teams—you name it. People just starting out will perhaps reap the biggest and most obvious benefit, but even if you've been in your career for a couple of decades, I think you'll find plenty of useful perspectives that will help you guide the rest of your career more effectively.

The contents are organized into a series of complementary topics, with each chapter essentially covering a single “soft skill” or similar subject. You can read them in any order, although I definitely suggest reading the first three chapters *first*, as those set the stage with some important definitions and shared concepts.

You're welcome to contact me on Twitter @concentratedDon, or via my website, DonJones.com, as well as through the official Manning forums.

liveBook discussion forum

Purchase of *Own Your Tech Career* includes free access to a private web forum run by Manning Publications where you can make comments about the book, ask technical questions, and receive help from the author and from other users. To access the forum, go to <https://livebook.manning.com/#!/book/own-your-tech-career/discussion>. You can also learn more about Manning's forums and the rules of conduct at <https://livebook.manning.com/#!/discussion>.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers and between readers and the author can take place. It is not a commitment to any specific amount of participation on the part of

the author, whose contribution to the forum remains voluntary (and unpaid). We suggest you try asking the author some challenging questions lest his interest stray! The forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

about the author

DON JONES has been a professional technologist since the mid-1990s. He is the author of dozens of tech books and has worked in companies large and small, from startup to established enterprise. Since the mid-2000s, he has run career development workshops to help tech professionals better align and manage their careers to serve their lives, goals, and dreams.

about the cover illustration

The figure on the cover of *Own Your Tech Career* is captioned “Homme Baschkir,” or Baschkir man. The illustration is taken from a collection of dress costumes from various countries by Jacques Grasset de Saint-Sauveur (1757–1810), titled *Costumes civils actuels de tous les peuples connus*, published in France in 1788. Each illustration is finely drawn and colored by hand. The rich variety of Grasset de Saint-Sauveur’s collection reminds us vividly of how culturally apart the world’s towns and regions were just 200 years ago. Isolated from each other, people spoke different dialects and languages. In the streets or in the countryside, it was easy to identify where they lived and what their trade or station in life was just by their dress.

The way we dress has changed since then and the diversity by region, so rich at the time, has faded away. It is now hard to tell apart the inhabitants of different continents, let alone different towns, regions, or countries. Perhaps we have traded cultural diversity for a more varied personal life—certainly for a more varied and fast-paced technological life.

At a time when it is hard to tell one computer book from another, Manning celebrates the inventiveness and initiative of the computer business with book covers based on the rich diversity of regional life of two centuries ago, brought back to life by Grasset de Saint-Sauveur’s pictures.

Introduction

Every technology professional needs two distinct, **overlapping, and complementary** skill sets in order to enjoy a successful career—no matter how you define success.

The first set of skills is your *hard skills*, the technology skills that let you get the job done every day. Those are your programming skills, your systems admin skills, your network engineering skills, your security skills, and so on. They’re the ones you likely focused on in school, and they’re probably the ones you think about most when it comes to looking and applying for jobs.

The second set of skills is what I call the *squishy* ones, or what the industry more commonly calls *soft skills*. These are the skills that require no technical prowess but that instead more often focus on human prowess. **These are skills like communication, teamwork, conflict resolution, leadership, and the like.**

In my 20-odd years in the tech world, I’ve found that the difference between the best technologists and the just-okay ones is the soft skills. The best technologists are amazing technical experts, of course, but they also engage with their peers, colleagues, customers, and coworkers more effectively. They bring a human side to technology, and it’s a big part of what drives their success.

Browse some job postings on your favorite job website, and you’ll find the usual smattering of technology desires: JavaScript, HTML, Linux, pen testing, Tableau, Cisco, you name it. But dig deeper, and **you’ll find that what companies seem to be really concerned about are these human skills, like teamwork and communications.** Those skills are often harder to evaluate in a job candidate, but they’re the key to making a team healthy and effective.

In this book, I present more than a dozen of these soft skills. I’ve been a technology professional almost my entire adult life, and so I discuss these skills as a

technology professional. Some of you will have already mastered some of these skills, and the layout of this book—one skill per chapter—should make it easy for you to skip those, if you want to. (Though there’s a good chance that I may offer a new perspective on the skills that are already part of your repertoire.)

I also present a handful of hard nontechnology skills that I feel are important for anyone to master in support of their career. These are hard skills, as in they are quantifiable and repeatable, but they are not related to technology—how to read a P&L statement so you can gauge the financial health of your company, for example, or how to read between the lines in a job posting. There are skills that have benefited my career a hundredfold over the years.

I’ve tried as much as possible to make everything in this book relevant to a global audience. I’m based in the United States, and so a few of the examples I offer are US-centric. Whenever that happens, I’ve tried to take care to acknowledge that limited perspective and offer what I can to help you explore a more localized perspective on your own. The underlying principles are universal, so hopefully those details that don’t apply in your country won’t deter you from picking up the underlying message.

I also want to acknowledge that the perspectives in this book are entirely my own. I’ve tried very hard, over the years, to let my perspectives evolve and be shaped by the many people I’ve worked with, but in the end I can really only offer you what I’ve gleaned from my own experiences. In no way do I want to present my perspectives as the objectively correct ones, let alone the only ones in the world. Instead, I hope you can simply absorb what I’ve written and fit them into your life wherever they offer you value. And please, don’t let this book be your only exploration into these important skills. We may call them soft, but that’s simply because they’re often hard to quantify and measure. They’re critical by any measure.

I hope you’ll feel free to reach out and let me know what you think of the book. You can always connect in the forums on [Manning.com](#), and that’s where any errata—typos and the like—should be noted, as the whole Manning team helps collect those for future printings. But you’re also welcome to contact me via the HMU page on my website, [DonJones.com](#), or even ping me on Twitter @concentratedDon (but please do keep the errata on [Manning.com](#) so it can be acted upon).

Thanks for reading.

DON JONES
Las Vegas, Nevada

Own your career



Many of us see the word *success* and think of a big salary or an important job title. Instead, *success* should be seen as a set of criteria that you define and that represent a career capable of supporting the personal life you want. A career plan is a plan to achieve that kind of career.

1.1 Job, career, success, and self

Let's begin by quickly defining some terms so that you and I can be on the same page. I define *job* as a set of tasks that form a role you're paid to do: a software developer, a data analyst, a systems administrator, a network engineer, a security specialist, and so on. A job is an arrangement in which the employer offers compensation to someone and that someone performs the requested job tasks. If you weren't doing the job, someone else would be. In other words, your employer *owns the job*. That means your employer bears a lot of responsibility for the job: they have to provide you the tools you need, they have to tell you which tasks need to be performed, and they get to define the standards that you have to adhere to when performing those tasks.

Your *career*, on the other hand, belongs to *you*. Your career encompasses all the skills it takes to obtain, keep, and perform the jobs of your choice, and you may have several different types of jobs over the course of your career. You are responsible for your career: you get to decide where it is headed, and you have to pay for any of its upkeep that falls outside the scope of your current job.

So, your employer owns your job; you own your career. Suppose that you're a software developer working on in-house applications written in C++. You've been doing this for a while, and you're eager for a change. You're also a little concerned

that being great in C++ doesn't offer many job opportunities in the world, and you're—wisely—worried about getting stuck in a rut by working in a language that's not too common.

You decide that you're really interested in web development, and you'd like to take a class in advanced JavaScript programming. You also want to attend a conference on web development so that you can get a handle on all the various technologies that come together in web applications. But your employer refuses to pay for either the class or the conference. Should you be upset?

I would argue that no, you should not. The class and the conference aren't connected to your *job*, meaning that neither the class nor the conference will do anything to make you a better C++ programmer, which is what your employer pays you for. Instead, the class is something you want for *your own career*. You want to expand the set of skills contained in your career, both to satisfy your own interest and to expand the job opportunities available to you. Therefore, you should be the one paying for the class and the conference, not your employer. But as you'll see in chapter 5, expanding your hard skills is an important aspect of maintaining your career. So don't cancel your class or conference just because you'll have to pay for them.

There's a downside to "owning" a technology career, in that it can be expensive. But there's a definite upside to owning a technology career too: you can make it serve your bidding. Your technology career can be a powerful means of achieving . . . well, just about anything you might want it to!

But what should you ask your career to do for you?

1.2 **Start at the beginning: With yourself**

Too many of us graduate from school, get into our first real job as an adult, and immediately start trying to do the best job we can. We manage to impress our employer, and in time, we're offered a promotion. Or perhaps we gain enough experience to land a better job elsewhere—one that pays more, offers a better job title, or has other upgrades.

Without really thinking about it, we start to equate success with salary, job title, the size of the team we lead, and other criteria. But we rarely stop to think, "What's it all for?" That's what I'd like you to do right now: stop and think, "What's it all for?"

What kind of life do you want to live? How do you want to spend your time, both on and off the job? What contributions do you want to make to the world? What passions and experiences do you want to pursue?

I want you not only to stop and think about these questions, but also write down your thoughts. Writing things on paper with a pen or pencil helps you take the thinking seriously—and helps you remember your conclusions.

Your answers to these questions clarify and define the life you desire, so I call this piece of writing your *life definition*. Unlike other definitions, your life definition may change as you enter new phases of life and discover new goals and values. My current life definition is not the same one I've always had. As I've grown older, formed

a family, and changed interests, my definition has changed. That's fine; that's what being alive is all about! But I'm careful to document, in writing, what I want from my life. I revisit this definition annually, and I treat it a bit like punching in a destination into a GPS app, in that this life definition is my destination. When I get there, I want to stay there, unless something happens that prompts me to rethink what I want from life.

One more thing before you start thinking and writing: I want you to write this life definition as though you were on the outside of your life looking in. Not to be depressing, but treat this definition as though it were a kind of extended obituary. It should represent what you want your life to look like when you're looking back on it. Writing it this way can help you distill your most important dreams, goals, and desires. To show you what I mean, I'll share my current life definition.

Don was an experienced technology educator who was respected by a broad community for his ability to help people learn new technologies. He was also an experienced business leader, and the companies he worked for relied on him to help execute their plans. They also relied on him to help build new leaders by nurturing his team members.

Don worked mainly for companies where he was able to make a meaningful and visible positive impact, even when that impact was visible only internally. He was a leader in the technology communities in which he participated. He helped build and grow communities that invited and encouraged others to participate, and he helped them see a path toward growing those communities on their own even after he bowed out.

Don was a well-regarded author of fantasy and science fiction. Although he never achieved Stephen King's level of success, his works were well reviewed and enjoyed by a meaningful-size audience.

He believed that technology skills were a key for people to lift themselves up in life, and he focused significant effort toward making those skills more accessible. He founded an enduring not-for-profit organization that unlocked tech skills for people in disadvantaged situations.

Don's family lived comfortably, if not luxuriously. They were able to maintain a small second home in a quiet place as a "retreat and recharge"

location. They traveled for vacations, visiting new countries and experiencing different cultures. Don ensured that in the event of an accident, his family would be well provided for.

This is what I want my life to look like when it's all over. I want you to take note of a few specific bits in this life definition:

- Money is implied in my definition because I've listed some things that will clearly require money to achieve. At this stage, however, I'm more interested in describing what I want than worrying about how much it will cost.
- I've described the types of jobs I want to take. Work is, after all, a big part of life: most of us spend a third of our lives working. It's important to me to have a job that I find satisfying and that provides the income I need.
- I've listed things that I haven't yet achieved, even though I'm pushing 50 years old. I may not ever achieve all my goals, but I'm pointing myself toward them.
- I've included some things about my personal life that will require time, which in turn creates certain implications for my work life. I need a job that will give me free time to work on writing novels, for example, which means that I probably won't be working for hard-charging startup companies that need 20-hour workdays.

My life definition is the destination that goes into my “GPS of life” app. Everything I do is meant to drive toward that destination. My life definition is what it's all for. It's why I get out of bed in the morning; it's why I work; it's why I live.

Please write it down . . . on paper

Human cognition—the way we think and learn—depends largely on our senses. The more senses are engaged in a given experience, the more lasting and powerful the resulting memories are inside our brains. That's why we can have such strong memories from our childhoods. The smells, the sounds, the sights, even the taste of the churro add up to that memorable moment at Disneyland.

Typing on a computer doesn't engage many of our senses. Our sense of touch and even our sense of sight barely register what we're doing: the words simply flow onto the screen, as they're meant to do.

But when you write down your definition of your life, you want your brain to be deeply and deliberately engaged. You want to be thoughtful, and you want the words you write to be branded into your mind. That way, those words will always be more present for you, helping you remember what it's all for.

That's why I've always written my life definitions over the years, using a sharp pencil and a quality notebook. Even sitting here typing this book's manuscript into a computer I can smell the leather of the notebook cover, feel and hear the scritch of the pencil

against the page, and feel my hand traveling across the page. Those sense memories instantly bring the key elements of my life definition to mind. I didn't even need to go get my journal to offer you the example I did, because my definition statements are indelibly carved into my consciousness. They're always top of mind—as they should be.

With my life definition—my GPS destination—firmly in front of me, I need to figure out what it's going to take to get there.

1.3 **What does success look like for you?**

For me, success is simple: it's whatever it will take to get the life I want. Success is literally a bulleted list of things I need to achieve that life I've envisioned and defined. If my life definition is what goes into the "GPS of life" app, my success is whatever parts I need to build a vehicle that will take me there.

Consequently, success is not something I pursue endlessly, but a specific set of measurable goals that I can slowly work toward. I will know when I reach those goals, and at that point, all I need to do is maintain that success rather than try to continue growing it. I never feel that I'm in a rat race, endlessly pursuing the next-bigger piece of cheese. Instead, I'm pursuing specific, achievable goals that will help me live the life I want.

When I write down my life definition, I also write down my success definition. To create my success definition, I usually start with the life definition and then add bullet points that describe what it will take to achieve my outcomes—aka my life definition. As much as possible, I try to keep the success bullets objective and measurable, so that anyone could look at my life and decide whether I'd met a given goal. That's not always possible; some things, especially more qualitative things, are always going to be a little subjective. That's fine; try to be as objective as reasonably possible. My idea of a great vacation, for example, isn't something that I can easily quantify; it even shifts a bit from year to year. I might be able to think only about ranges or guidelines, and that's fine. Here are some examples of my success definition, using a portion of my goals from my life definition:

Don was also an experienced business leader, and the companies he worked for relied on him to help execute their plans. They also relied on him to help build new leaders by nurturing his team members.

* At least a senior director or vice president role

* A team that includes people who have their own direct reports

* A company with a known habit of promoting from within

Don was a well-regarded author of fantasy and science fiction. Although he never achieved Stephen King's level of success, his works were well reviewed and were enjoyed by a meaningful-size audience.

- * Job with solid work-life balance—no weekends and no late nights
- * Don't feel comfortable with writing dialogue; need to publish a short novel that doesn't involve characters speaking to one another
- * Intend to self-publish; need to understand the marketing aspects and budget accordingly

He believed that technology skills were a key for people to lift themselves up in life, and he focused significant effort toward making those skills more accessible. He founded an enduring not-for-profit organization that helped unlocked tech skills for people in disadvantaged situations.

- * Need to understand how to found a not-for-profit in the United States
- * Need to create a way for the tech community to derive value from the not-for-profit in exchange for money, so that the organization has an operating budget and can execute on its charitable mission
 - o Launch a tech conference
 - o Profits from tech conference power the organization
 - o Partner with existing tech education not-for-profits to reach a client base

Don's family lived comfortably, if not luxuriously. They were able to maintain a small second home in a quiet place as a "retreat and recharge" location. They traveled for vacations, visiting new countries and experiencing different cultures. Don ensured that in the event of an accident, his family would be well provided for.

- * Need \$150,000 per year gross
 - o Includes reasonable mortgage on a cabin near Dixie National Forest (three-hour drive)
 - o Includes long-term disability insurance

- o Includes term life insurance
- o Includes retirement contributions per financial adviser's suggestions
- o Includes annual vacation budget

This example is pared down, but I wanted to highlight some sections that are a little bit subjective, as well as some (such as the salary calculation, which is a made-up number for this book) that are objective and easily measured. Some items are definitely aspirational, as they're things that I hadn't yet achieved when I wrote them down.

It's important to note that I don't change the success bullets unless I'm changing the life definition or unless life itself has changed around me. I don't plug in a larger salary number for no reason, for example. Instead, if I'm feeling that I need more money from work, I look at *why*. Has the cost of living gone up? Did we decide to adopt a kid? Have we been visiting more-expensive places than I anticipated? Whatever the reason, I need to decide what has to change in my life definition to justify the need for a larger salary. Have we been eating out more than we should and spending more on food as a result? Maybe we should decide to stop that. Or if eating out is something that we love and want to continue, I need to modify my life definition accordingly.

The point is that my success is there to support my life. Nothing goes into my success definition unless I know why it's there, which ensures that I'm not randomly chasing a larger salary or pointlessly pursuing a job title for the glory of it. Whatever I'm doing with my career, I'm doing it because my life needs it. With those success bullets in mind, I can start crafting a career plan.

1.4 **Creating a career plan for right now**

If your life definition is the destination you plug into your GPS, and if your success targets are the parts of the vehicle that will get you there, your career plan is the route that the GPS spits out. Follow that route, and you'll get there.

With a good career plan, however, you don't need to know the entire route at the outset. Instead, you can think about the next few steps on the journey and plot them out for yourself. So long as you're always pointing to your ultimate destination—your life definition—you'll get there eventually.

To create my career plan, I start by looking at the success bullets that support the components of my life definition. Some of the success targets might be too hard for me to imagine reaching. Become a company vice president at 25 years of age? Hah! Instead, I focus on the bullets I can achieve or at least see a path to. Rewinding my mind over two decades (and consulting the notebooks I wrote everything in), I see things like this:

Don was also an experienced business leader, and the companies he worked for relied on him to help execute their plans. They also relied on him to help build new leaders by nurturing his team members.

- * At least a senior director or vice president role
- * A team that includes people who have their own direct reports
- * A company with a known habit of promoting from within

Okay, at 30 years old, I definitely lacked the experience to get a role as senior director or vice president. But by then, I'd led a small team, and I'd started to understand what running a business was all about. So I made a point to land a director role, specifically with a company that would be interested in investing in making me a better business leader. With that as a job-hunt focus, I was able to find a role within my current company at the time. The pay wasn't as much of an increase as I'd wanted, but I was more focused on getting the experience I needed to move toward my goals.

And that's what a career plan is: a way to move toward your goals, to slowly tick off all those success bullets. Focus on the ones you can achieve now, and for the others, start researching what it would take to achieve them. I spent quite some time on job boards, for example, looking at what was being asked of candidates for vice president jobs. I knew that I'd need credentials such as these:

- 10 or more years managing teams of however many people
- More than five years in a position with financial accountability or outcome accountability
- Ability to manage teams with more than four layers of people
- Experience managing up into the executive suite

I didn't have those qualifications, but I could start to see the path toward them: start with a small team, ask my boss to share accountability with me, ask to give an occasional presentation to the company executives. Again, meeting all these goals required working for someone who was willing to make those investments in me, and my job hunting was aimed sharply in that direction.

Scouring job postings is a wonderful way to build a career plan. At one stage of my life, I realized that my next move needed to be an increase in salary—not a huge one, but I was pretty much maxed out in terms of where my current job would take me. In fact, I came to realize that the field I was in was unlikely to offer more money, so I needed to change fields, moving from systems administration to software development. Boy, was that scary. I needed to pay for the necessary education myself because my then-current employer had no reason to pay for it. But I did it. I eventually got a job as

lead web developer and wound up in charge of a small team, which also put me on track toward my leadership goals.

Obviously, the process took a while. Life can be a long journey. But at age 46, I finally did land my vice president role. Career plans work.

1.5 Action items

In each chapter of this book, I offer you an exercise. I strongly recommend that you do these exercises, because each one will help you start mastering the soft skill covered in that chapter.

For this chapter, as you can probably expect, I'm going to ask you to get a paper notebook and a pencil (unless you love pens, which is fine), and write down your definitions of life and success, as well as make a first pass at a career plan. Work with your family and friends on this task; they're stakeholders in your life, and they need to be represented in your life definition:

- Start by writing your life definition as you see it. Given who you are and where you are in your life right now, what can you imagine doing? If you find it difficult to think about this definition as a set of end-of-life retrospectives, then instead, focus on what you, at this point, find valuable about your life and what you want in that life for the foreseeable future. If you have clear goals, write them down as well.
- Go on to your success bullets. What will it take, from your career, to meet your life goals? Get serious about the math in the appropriate spots; know how much money you'll need to make to get what you want. If the number seems to be unattainable, go back to your life definition and decide what you can live without, but try to make the life/money equation balance. In other words, for whatever you've said you want from life, make sure that you provide a good estimate of how much money that will take.
- Finally, start thinking about the single next step in your career plan. What one, two, or few things could you change about your career to achieve one or two of those success bullets, such as a better-paying job? Do some research, figure out how to take the next step, and then start executing that step.



Build and maintain your brand

When you think of a brand you know and love—Coca-Cola, perhaps, or Disney, or any other major consumer brand—certain expectations come to mind. You love a brand because those expectations exist and are consistently met. So what expectations do you create (or want to create) in other people and consistently meet? Are they positive expectations that help unlock career opportunities? What, in other words, is your brand?

Your brand, quite simply, tells a potential employer who you are. It tells them what to expect from you. They build that brand from everything they can see and learn about you: from personal interactions or hearsay to social media, open source projects, and Q&A websites where you participate—basically, everything online and everything offline as well. Your brand helps an employer see what you'll bring into the workplace, so you should ensure that your brand reflects what you genuinely will bring.

2.1 *Brand building: Know your audience*

Whether you like it or not, and whether you take an active hand in it or not, you have a personal brand. Just the way you dress for a job interview is part of that brand.

The marketing departments of large corporations spend a considerable amount of time defining their audience so they can tailor their marketing efforts to be most effective. Gatorade, for example, probably works to impress a completely different audience from, say, Bacardi. Understanding who the brand must speak to—really understanding who they are and what they care about—is the key to building a brand that speaks to its audience.

With products like beverages, brand-building usually starts with identifying and learning about the audience. Products exist to fill a need, and that need is defined by the audience. Who are we trying to sell products to? The entire development of the product and its brand are driven by that audience: when it comes to our product, we're going to do only the things that resonate with our intended brand. Our research shows that athletes are drawn to bright-colored drinks, for some reason, so we're going to create beverages in fluorescent colors. Kids like candy flavors; older people may respond better to nostalgia, and so on. The drivers of the audience help craft the brand and the product that the brand represents.

That's what *brand* means: a set of expectations that anyone interacting with the brand has. People like a brand and even become loyal to it because of those expectations and because the underlying product meets those expectations.

You have a brand too. It's your personal brand, and like a corporate brand, it tells people—your employer, your co-workers, and your colleagues—what to expect from you. This brand is visible even before you meet people thanks to your interactions on social media, in open source projects, and other online locations.

With your brand, you can't necessarily change who you are, so understanding your audience is less about building your product. You already are the product. But you still need to understand what's important to your audience. Let's look at some hypothetical examples to show you what I mean by "understanding your audience."

PUTTING MY EXAMPLES IN CONTEXT The following examples are built on stereotypes of industries. I don't in any way mean to imply that these industries are like these examples; I want to use straightforward examples to demonstrate the concept of understanding your audience.

Consider a bank: a traditional, 200-year-old company. We're talking about a large national or multinational banking corporation, the kind whose executives wear three-piece suits and whose office buildings are towering edifices. When considering technology professionals, what might a company like this value? A trim, neat appearance in the office, perhaps. A fairly conservative person, maybe, who is dependable and doesn't take a lot of unnecessary risks. Someone who's punctual and who understands the need for information to be safe and secure. Someone who's comfortable in meetings, because big banks have lots of divisions that need to coordinate. Maybe someone who's comfortable using older, proven technologies.

Now consider a brand-new, lean technology startup with one office and a handful of employees. What might this company value in tech professionals? Perhaps a willingness to work long hours every day of the week. Someone who's a little quirky, maybe, who can think outside the box. Maybe someone who's well-known in their portion of the industry and is thought of as an innovator and leader.

As you can see, it's important to consider who your audience might be and what's valuable to them. You don't have to pick one audience, and you don't even have to choose one industry. Your brand can appeal to a variety of audiences, if you do things

right. But you need to understand what's valuable to the companies and people to whom you hope to appeal.

How would I go about branding myself to appeal to both the large, established bank and the small, agile startup? What value do I bring to the table, and how can I communicate that value succinctly in a brand?

- I might ensure that my visible brand is **highly focused on technology**. I might tone down the more personal aspects of my visible brand, such as where I like to vacation and what politicians I support. Neither company is likely to care about those things not being in my brand, but they might be turned off if those things were part of my brand.
- I would probably ensure that I **had a lot of community contributions—blog articles, open source projects, and other visible contributions**—that showed **my work**. At the same time, I'd be careful to focus on the security aspect of my **work**. I wouldn't post code in open repositories and expose API keys, for example. **Demonstrating that I follow best practices for security would appeal to any company**.
- I'd make sure that my appearance—**my social media avatars and the like**—reflects **a neat, businesslike appearance**. Pink hair might be acceptable at a startup but maybe not at a bank, and the startup probably won't mind if my hair is a natural color.

These ideas reflect a specific approach that I'm taking. **I'm saying that I want to be appealing, as a brand, to a couple of different audiences**. That means I'm going to have **to find a common denominator and go with that as my personal brand**. But there are also some things I explicitly won't do, right? Let's look at those things in table form.

Branding item	Definitely do	Definitely avoid
Visible brand (social media and the like)	Focus on technology or whatever is applicable to my career, activities that an employer would expect to see me doing in the workplace	Discuss contentious topics such as politics, which might suggest that I would bring those topics into the workplace with me
Contributions and work examples	Ensure that I'm seen as a community contributor through blog posts, code submissions, and so on	Allow those public contributions to reflect bad practices in security and privacy
Physical appearance (social media avatars and so on)	Ensure that it's neat and business-like, showing the real me	Project an image that is not appropriate for the industries I intend to work in

You might feel entirely differently. You might want to work only for aggressive, exciting startups, which means you could tailor your brand more precisely to what that audience values. What you decide is okay, and what's not okay, for your brand is your decision. Please don't see my examples as being some kind of directive; they're decisions I made for myself, not for you.

The point is that your brand will say different things to different people. To some people, Coke says, “Refreshing, energizing beverage that I enjoy.” To other people, that same brand says, “Sugary junk food that should be banned from the planet.” Coke obviously focuses its efforts on the first group of people and doesn’t worry too much about the other group, at least in terms of how it presents their brand. So you need to decide who your brand will appeal to, and you may want to tweak your brand if it doesn’t appeal to the companies you want to work for. By defining and understanding your audience, you’ll know what’s important to them, and you will be able to market yourself as a professional who helps meet their needs.

IT'S STILL OKAY TO BE YOURSELF I added the comment about having pink hair on purpose. Please don’t feel compelled to take something out of your brand if it represents who you really are. Your brand reflects what you bring to the workplace, and it should be authentic. Just be aware that some employers may not like your brand. You can’t please all the people all the time, right? But if having a certain hairstyle is a genuine part of you, anyone who’s turned off by it probably doesn’t deserve you in the first place, so there’s no need to edit that part of yourself. I would look silly with pink hair, but you might rock it. That’s why my specific brand decisions won’t be yours.

2.2 Social media and your brand

Social media is a tremendous part of your personal brand these days. Whether you’re looking for a job or already have one, what happens in your publicly accessible social media accounts matters. Your activity on social media informs your co-workers, your employer, and your potential employers who you really are. People tend to be at their most candid on social media, so everyone else more or less takes you at your word. To the world, what you put on social media is who you are.

I use Facebook sparingly, mainly to connect with, and share news and photos with, a dozen or so friends who are scattered across the country. I can be my real self with those people because I’ve known most of them for most of my life. They’re the ones who can see the photos of that time in Mexico when I got a little carried away at the cantina. As a result, my Facebook profile is incredibly locked down. My friends can’t even tag me in a post unless I approve it. Nobody I work with is a “friend” on Facebook unless we’ve actually spent personal time together. Facebook is not part of my personal brand; it’s not something that the public gets to peruse.

Twitter, on the other hand, is very much part of my professional life. So is my blog. Those places are where I engage with my professional audience. The content I post in those places reflects my work life, my community involvement, and the work I do. I’m absolutely fine with my current employer or a potential employer seeing my Twitter feed (@concentratedDon) or my blog (DonJones.com). Even the political content of my blog is safe for work, because it’s mainly explaining how various aspects of the US political system work, as opposed to rants about my political opinions or attacks on political opponents. Even my personal comments on Twitter are the kind of innocuous

chat that almost anyone might find acceptable around the water cooler in the office. I try very hard to ensure that both Twitter and my blog reflect my brand—the person you can expect me to be at work.

Be you in your brand

If you're taking the time to actively manage your public brand—and I think you should—make sure that it visibly represents you. Specifically, make sure your social media avatar (the little picture that appears next to your postings) is a picture of you—not your cat, not your family, not an abstract geometric pattern, and not the logo of your favorite superhero. You are your brand, not those other things.

Use the same photo everywhere your public brand exists. This photo ties all your appearances together and helps the people experiencing your brand recognize that they are pieces of the same whole.

Let's say that you're about to upload your photo, and you're thinking, "Well, I'd really rather use a picture of my kid here to show her how proud I am of her and how much I love her." **That sounds like part of your personal life, not part of the brand you present to the world.** Try to differentiate between those two things. Use that avatar on your closed, private social media accounts only.

Use your real name, too. Doing so indicates that you're being real, not trying to hide behind a pseudonym or handle.

My deliberate separation of the personal and the professional does not mean that I'm plain-vanilla online in an effort to appeal to everyone. Ask anyone who's seen me speak at a conference or teach a class, and they'll tell you that good-natured sarcasm is very much part of my brand. I use it tactically to help make important points stick, and for my audience (the people I'm engaging with as part of my work), it mainly works. I do recognize that some people don't like that approach, and as a result, they don't really relate to my brand. That's fine. You can't be all things to all people, and I accept that whatever appeal I might have isn't universal. But I've taken the time to know my audience, I accept what works for them, and I'm okay with making those things part of my brand. That doesn't make me a win for every audience, and I'm okay with that too.

What does your public social media footprint say about your brand? If it says nothing, people will simply infer whatever they want from whatever evidence they see. Not managing your brand isn't the same as not having a brand; as I have said, **we all have a brand.** **If your brand seems to say nothing, people might conclude that you're not publicly engaged.** For some audiences, that conclusion might be fine; for others, it might not resonate in a positive way.

My social media brand is communicated largely through LinkedIn, Twitter, my blog, my professionally focused YouTube channel, and the writing I've done for a huge variety of technology websites. I essentially have no other social media footprint, because anything else I do is massively locked off from the rest of the world, accessible only to a small group of friends and family.

I keep my personal Facebook account cut off from my public image, but I do not cut off social media as a whole. I use it actively. I use LinkedIn because it's the de facto social media network of technology professionals. I use Twitter because I have an audience that engages with me there. I use YouTube because it's an opportunity to use my videos to reinforce what my brand is all about. If a potential employer or my current one wants to get to know Don, they'll find plenty to look at—largely on-brand. I wouldn't want someone to start looking me up and find nothing, because then I wouldn't be able to manage my brand. If I weren't publicly active on social media, I'd be left to whatever an employer's imagination provides, which isn't what I want.

I also need to be aware of what I repeat or share in public social media. Want to know what's in my Twitter feed? Mostly news from the many Disney Parks websites I read, along with a smattering of Apple rumors. That's what I like to read, and it's what I've taught Twitter to show me. I also have a few comedians in the mix, who can get pretty snarky when it comes to the political events of the day. I don't necessarily agree with all of their opinions, but I do enjoy reading them. That said, I don't retweet them. Even if I think my audience might find those opinions interesting to read and discuss, I don't want to imply that I agree with those opinions (I don't always, but they can be interesting without my agreement) or that my Twitter content is straying from its path. I don't want to take a risk that a potential client or employer might see me as the kind of person who brings controversy into the workplace. So I'll ask again: What does your public social media footprint say about your brand? Does it say things that would appeal to your target audience or not?

NOTE It's always a good idea to understand what your brand is saying about you, especially on social media. It's critical that you do so before a job search, because many organizations will consider your public brand—as communicated on social media—in their decision-making processes.

2.3 Your brand has a wide reach

Never forget that we live in a small world. I once worked with a fellow named Mark. He was a great guy, and we got along really well. He was one of the few professional acquaintances I've made who truly saw me at my worst on occasion—like the time the server migration we worked on did not go smoothly. In time, I left the company where we worked; he stayed on longer but also left eventually.

Some years later, I was applying for a job with a new company. I got through to an interview, and my prospective new boss started with “So I understand you're a big fan of Disney Parks! Which is your favorite?”

Mind you, this happened long before the advent of social media. There was no way for a complete stranger to know that I liked Disney Parks. But Mark already worked for them. He'd started with them a few months earlier, had recommended me for the job, and had told them a bit about me.

Fortunately, I'd never given Mark too many negative experiences to talk about. Our work interactions had been positive, and we saw each other as solid professionals. But it

struck me: What if I'd really been a jerk to work with? Singlehandedly, Mark could have killed my ability to get that job. I'd probably never have gotten an interview. Our relationship—positive or negative—would have affected whether I got the job.

The world is much smaller today. Your brand reach is vastly greater than you may realize. People you've never met have definitely heard about you or could hear more about you with very little effort. In a lot of ways, the expanded reach of your brand today is a good thing. It's what will get you that next job or other opportunity, but it can obviously work against you as well.

The take-home message is to spend every moment of every working day pretending that it's the job interview for your next promotion, job, contract, or other opportunity. Everything you do today will have an impact on how you're perceived in the future. It's important to be a consummate professional all the time because your personal brand will be a positive reflection of you as a technology professional.

USE THE REACH OF YOUR BRAND In chapter 3, “Network,” I explore the idea of extending the reach of your brand by continually using networking as part of your career. And in chapter 4, “Be part of a community,” I look at how being a positive community contributor can add serious panache to your personal brand.

2.4

Professionalism and your brand

Do you know the biggest concern every hiring manager has when they decide to extend an offer to a job candidate? It's whether that person is a professional—someone who can work in a team environment with other human beings; someone who can be polite, efficient, and effective; someone who can make the workplace better, not worse.

Those things are hard to pull out in an interview. That's why recruiters and hiring managers will search for you on social media and seek references from other people in the field—not just the references you provide, which are always upbeat, but real references—co-workers they've located on LinkedIn, for example, or people they've found through a thorough review of your social media footprint. Before they make you an offer, they want to get to know you.

Professionalism is an important part of your brand. You want to be known as someone who shows up at work every day to get the job done; someone who can work with other people, accepting that they will have different opinions, backgrounds, and cultures; someone who can manage their time effectively; someone who keeps their word, pays attention to details, and can be a supportive member of a team.

KNOW THE CHARACTERISTICS OF SUCCESSFUL TECH PROFESSIONALS In chapter 6, “Show up as a professional,” I explore several key behaviors that can help support and build a solid brand.

You need to not only exhibit professionalism, but also find a way to make it part of your brand. Do you have a blog? Don't feel that you always need to write about

technology topics! Take a break now and then to write about professionalism, or an aspect of it, and what it means to you. Engage on social media on the topic. Offer suggestions for time management techniques, or share how you've struggled with interpersonal communications at work. Your brand is how you're seen, and you want to make sure that you're seen as a person who thinks about professionalism and is deliberate about it. You might be surprised by the opportunities that this focus on professionalism unlocks for you!

2.5 How to sabotage your brand

There are lots of ways to sabotage your brand. Some of them you can probably guess from reading this chapter, such as being inappropriate on social media, behaving unprofessionally, or misunderstanding your brand audience. Becoming known as a poor manager of your own time is another way. Being thought of as lazy is another; you may not be lazy, but if that reputation attaches to you, it can damage your brand. If you work remotely some or all of the time, it is a mark against your brand to be seen as a poor remote worker—someone who goofs off without supervision, is not available during office hours, or doesn't answer calls or return emails promptly.

The simple-to-remember, hard-to-execute basic truth is this: *everything you do at work, or in front of your colleagues or employer, affects your brand*. Whether you like it or not, you do have a personal brand. All of us have one. Our brand is simply the way other people perceive us and think about us. We can take control of that brand, and work to make it (and keep it) a positive, career-enhancing thing, or we can simply let the chips fall where they may.

Everything you do within sight of your colleagues and employer affects your brand. That “within sight” piece has become much bigger in the past decade, with the rise of social media and our always-connected lives. It may seem unfair that an employer would penalize you for something you posted on Facebook or that you didn't get a new job because word got around that you were perceived as lazy. That's the world we live in, though, which is why it's so important to actively manage that personal brand of yours.

2.6 Further reading

- *Introduction to Personal Branding: 10 Steps Toward a New Professional You*, Mel Carson (independently published, 2016)
- *Branding Pays: The Five-Step System to Reinvent Your Personal Brand*, Karen Kang (BrandingPays Media, 2013)
- *LinkedIn for Personal Branding: The Ultimate Guide*, Sandra Long (independently published, 2020)

2.7 Action items

For this chapter, I'd like you to evaluate your brand. We all project a brand, whether we intend to or not, and by evaluating its current state, we can decide whether we want to make any changes and what those changes might be. For this exercise, consider the following:

- **Where does my brand exist?** Include in-person encounters (like those you have at the office), as well as social media, technology communities in which you participate, and other online presences.
- **What does my brand say?** As developers know, *empty* is not the same as *null*; your brand says something, even if only "This person doesn't seem to engage much." **To find out what impression you are making on other people, you can ask colleagues and co-workers what expectations they have of you, based on your performance. How do they expect you to handle a project, collaborate on a team, and behave on the job?** Set up an anonymous survey if that helps them be more candid with you. **Ask the people who encounter you online what they expect from you, based on what you post.** A friend who performed this exercise was alarmed to learn that many of his Facebook friends considered him to be a person who was primarily interested in politics. Politics was the main content he reshared on the platform, so it was the main image of his brand there.
- **What about my brand contributes to my success definition?** Go back to your success bullets from chapter 1. What aspects of your current brand support some of those bullets? Does anything about your current brand detract from any of those bullets?
- **What could my brand do to better support my success?** Consider all the places where your brand exists. If part of your life definition is to be a valued contributor to a technical community, are you doing that? Can your contributions be seen in a way that defines your brand as a contributor?

When you're done, look at the results and decide whether your brand could use a more deliberate approach. What might you do differently to create a brand that more perfectly aligns with your success definition? Do you need to amend your success definition to include brand-building and brand-maintenance goals?

3 Network

In our always-connected world, it's sometimes easy to lose track of the value of direct human interaction. But that interaction—networking, in other words—is one of the most valuable aspects of any career, and it's a critical soft skill to master and maintain.

3.1 Why networking?

Networking is the process of getting to know other people in your field. I'll share a sad fact from recent experience: you can upload your résumé to as many online job postings as you want, and you're likely to hear back from only a tiny fraction of them, no matter how well qualified you are. Even with the best résumé, finely tuned for the artificial intelligence algorithms that take a first pass on all applicants, the odds that a human being will ever see your information are slim. Many online job postings get thousands of applications—too many for any one to stand out easily.

In writing this book, I spoke with hundreds of people who got new tech jobs, moved up in their companies, or transferred to different teams in their companies. The majority of these people told me that they'd never have gotten the job if someone else hadn't spoken up for them internally. If their network of colleagues hadn't lifted their résumé out of the digital pile and called the hiring manager's attention to it, they never would have gotten an interview. *That's "why networking."*

Over the past two decades, my network of colleagues has helped me land new customers, and get long-running magazine columns, book contracts, new jobs, and speaking engagements at conferences. Some of my projects would have been impossible to start without my network's help, such as my book *Shell of an Idea: the Untold History of PowerShell* (independently published, 2020). I would not be exaggerating to

say that nearly every good thing that has happened to me professionally has happened with the help of my network.

I've worked hard to build a professional brand (as discussed in chapter 2), and my network is the group of people who know that brand. Because I try to stay consistent with that brand, my network knows what to expect of me. They know what I'm capable of, and they have a good idea of the value I bring. I do the same for them; our relationship is very much a two-way street. I've been happy to help the people in my network make new connections, get more eyeballs on their projects, and even land new jobs.

But here's the thing: networks take a long time to build and require constant engagement to maintain. You cannot decide to start networking a week before you need that network to help you. The saying "You must dig the well before you're thirsty" is never more appropriate than when it's applied to your professional network. To build that network, you need to

- Be visible within your part of the technology field
- Be known to people in the field
- Be seen as a valuable contributor to the field, even in small ways

This chapter focuses on some tips and techniques for achieving those goals.

3.2 *The problem with digital communications*

Technology professionals are obviously comfortable working with technology, right? Many of today's biggest companies rely on technology for even the most minute day-to-day communications. We message one another in Slack or Microsoft Teams; send emails between organizations; and even communicate with friends and family through text messages, Facebook, and other digital means. With so many of us working remotely, those digital channels are crucial to our work and our lives.

But digital communications aren't a natural way for the human brain to communicate. One of the biggest bane of remote workers, for example, is missing out on those hallway conversations and water-cooler discussions that happen constantly and organically in the office. We're people: we bump into others as we're walking around, we take a quick trip to someone else's desk to discuss something, and we share information over lunch or coffee.

Digital communications often fail to make an impression. As I described in chapter 2, our brains form stronger and more lasting memories when multiple senses are engaged. When we're reading a text message, we have at best one sense engaged: sight. Our brains aren't getting all the sensory input from an in-person reaction, which may include sight, sound, and smells. Our brains aren't getting the body language of the person we're communicating with, and body language is a tremendously important part of our total communications capabilities, as well as a critical part of how our brains perceive other humans.

It is possible to do some networking online, such as through social media. Even the back-and-forth that contributors engage in on GitHub repositories is a form of

networking. But that networking lacks the impact of in-person networking, so you need to engage in a lot more of it, and do so more continuously, to make the impact you want.

My point is simple: don't rely entirely on digital communications for networking. Some of your networking needs to be done in person, or at the very least via group video calls. Networking this way will probably require an investment of your time and possibly your money, but it is essential for building a network that will be there when you need it. It's also essential for giving you the opportunity to be there for others when they need you.

3.3 Ideas for in-person networking

In-person networking can happen on scales both large and small. If you tend toward the introvert side of the personality spectrum, smaller events—even those you organize yourself—can be more comfortable places to start. Then you can work up to larger events. Here are some ideas to get you started:

- *Network within your own company.* Go cross-team and find others in your role or similar roles—all the frontend web developers in the company, for example—and set up guild meetings where everyone can meet, introduce themselves, and discuss what they're working on in an informal, after-hours setting.
- *Look for local user groups that are related to your field.* It can be tough to take even more time after work to do business, but commit to it. Make sure that the meetings don't consist of everyone sitting and listening to a lecture; you want time to meet and talk with the other group members.
- *Consider regional conference-style events.* The Microsoft SQL Server community, for example, has a robust schedule of SQL Saturday events. These events are often inexpensive and one day long. They're a great place to network.
- *Attend small and medium-size conferences.* Medium-size conferences—often produced by a staff of volunteers or by media companies—offer a less-expensive and less-intimidating alternative to the 20,000-person trade shows put on by major technology vendors. Smaller conferences tend to offer more, and more approachable, networking opportunities as well. Community-run events (typically produced by a staff of volunteers) often have a friendly group of returning attendees, who can make the event more approachable for newcomers.

Remember that you can take the reins by helping to schedule small local events yourself. Is there no user group in your area? Start one! Local libraries may offer free meeting space, or local technology training companies may offer their classrooms free of charge when classes are over for the day. Advertise on Twitter, LinkedIn, and other social media. You may have only a handful of people to begin with, but if you stick with it, you can help create something powerful and helpful—a pretty great impression to make on those people!

3.4 Ideas for online networking

As I stated previously in this chapter, online networking is valid and necessary, but it's harder and requires more continual effort than in-person networking. I try to practice both types because they tend to attract different people, which adds diversity to the network of people I know.

Online events seek to reproduce the conditions and impact of in-person events:

- Online (or virtual) user group meetings often feature a guest presenter and are a great way to learn something new in an hour or so. Not all virtual user groups focus on networking, however. Try to find one that offers opportunities for people to break out into smaller groups, chat for a bit, and get to know one another. Or work with a virtual user group that is willing to add breakout opportunities. Online meeting services such as Zoom often offer virtual breakout rooms for this purpose, and can even assign attendees to a room randomly, which is a great way to reconnect with old friends and make new ones at the same time.
- Online conferences are great learning opportunities, but like virtual user groups, they often focus more on delivering information and less on networking. Look for ones that offer some kind of networking opportunities where you can meet new people.

Websites like Meetup.com, which is heavily used by tech groups, can be a great way to discover both in-person and online events that might be of interest. I will note that I don't consider venues such as LinkedIn and Faceplace to be great networking opportunities. They're good for communicating with an audience, but you can't "network" in 250 characters while the whole world is watching. Networking isn't done in bulk, and it isn't done in snippets. Social media can be great for helping to build and maintain a brand (as discussed in chapter 2), but it isn't a way to form strong professional connections.

Events aren't the only way to network online, though, and they're not even the widest-reach way you can network without leaving home. Here are some more ideas:

- *Commit yourself to a question-and-answer website that serves your field.* This site might be a general tech site like StackOverflow.com or something more topic-specific that's run by a volunteer group. Become known as someone who provides friendly, helpful, accurate answers; never tell people, "You can Google that." Check the site daily so that you can respond in a timely manner. Sometimes, even if a question has been answered, you can provide additional context or explanation, alternative solutions, or background information to help people.
- *Become a contributor to an open source software project that's meaningful to you.* If you use an open source framework, why not offer to help do code reviews, proof-read or expand documentation, or make some other contribution? You don't necessarily have to invent new code to become a helpful part of that software's community.

- *Follow key hashtags on social media.* I follow #PowerShell, for example. Following hashtags is a great way to find people who need help and to become known for the help you provide.
- *Start a blog.* Blogging is a well-known way to contribute to a technical community, but you need to be able to commit to blogging consistently. I aim to blog once a week and sometimes take a day to sit down and write an entire month's worth of articles. Just because someone else has already blogged on a topic you'd like to write about doesn't mean you shouldn't do it. Your unique perspective is likely to help people whom other authors weren't able to reach. Publicize your blog posts on social media (most good blogging platforms, such as [WordPress.com](#) and [Medium.com](#), can automate this for you) to start gaining a readership. Just note that blogging is a one-way thing: you're not truly networking. You're developing an audience of people who know you, though, and who see your brand, so it's still a valuable part of an overall networking strategy.

BE SEEN AS YOURSELF ONLINE It's important for your online presence to include your real photo and your real name as much as possible. Otherwise, you're not truly networking because people aren't getting to know *you*. That's Branding 101, as discussed in chapter 2.

3.5 *Etiqutte for networking*

Whenever and wherever you're networking, the people you're engaging with become part of the audience for your professional brand. You obviously want them to have a positive experience when they engage with you. Here are some tips for keeping your interactions brand-positive.

3.5.1 *In person*

In-person networking can be one of the most impactful forms of networking you engage in because it employs all of our human behaviors, including body language, facial expressions, and tone of voice. Meeting someone in person creates a far stronger mental impact than meeting them online, even in a video chat, so it's important to make a great first impression.

PHYSICAL APPEARANCE

Pay attention to your physical appearance, and make sure that it's fully appropriate for whatever activity you're participating in. At a formal business mixer, professional attire and a neat, well-groomed look might be best. For an informal meetup of geeks who jog on weekends, you're going to be wearing running gear. Whatever the occasion, make sure that you're focusing on the expectations of the situation.

Even professional attire will differ based on when and where you live and work. Where I live (US West Coast), business-lunch attire for a software engineer at a large corporation might include khakis and a golf shirt. But that same person, if they worked for a large East Coast bank, might expect to wear a business suit. At Microsoft's

famously informal campus in the Pacific Northwest, I've had business lunches with people in T-shirts, shorts, and flip-flops, and that attire was totally fine with everyone. Take the time to look around and see what other folks in your area, and in your line of business, are wearing; that's the best way to understand the social norms of your situation.

BODY LANGUAGE

Focus on your body language, and practice it with family and friends. Greet people with a smile and (if appropriate) a firm handshake. Stand with your arms at your side, not in your pockets or crossed in front of you. Maintain eye contact with whom-ever you're speaking to, and if you're in a small group, rotate eye contact among people. Keep your posture upright when you're standing, without seeming stiff. All these little gestures and postures add up to an impression that you are engaged and paying attention.

ICEBREAKERS, OR HOW TO START A CONVERSATION

Selecting appropriate topics of conversation at an in-person networking opportunity can be incredibly important. I like to avoid telling jokes, in large part because I'm terrible at them, and there's a real risk of offending someone. Also, most of the jokes I know are . . . let's just say less appropriate for a professional crowd.

But you still need icebreakers, or ways to bring yourself into a small group conversation. One of my icebreakers is to tell a short anecdote about an incident I had at work that others in the group might relate to. I prepare these anecdotes and even practice them, telling them to my friends. I try to have a few stories about a time when I failed at something, and I make sure to include what I learned from the situation. When others have shared these kinds of fails with me in conversation, those stories humanized them for me, and I wanted to try doing the same for myself. A while back, I worked in database administration, and if I found myself in a crowd of fellow database admins, I might have brought up a story like this:

So you mentioned optimizing. I was working with a real estate firm for a while, and we had this enormous database with all of our property listings. We're talking millions of properties across the world, including every property we'd ever listed. As we were designing the tables, we decided to fully normalize. But it turns out addresses are really complex, right? You've got the street number and name, but you've also got a prename direction, like North or South; potentially a lastname direction; the road designation, like "Road" or "Avenue" or something. And all that. It turned out to be 11 fields! [People's eyes got a bit wide as they anticipated where this story was going.] So yeah, every address lookup had to join nine tables, because we normalized the compass directions, the list of "Street" and "Avenue" designations, and everything else. Performance was terrible. That's where I learned about denormalization. [I tried to look sheepish.]

If you're getting started with in-person networking, it's fine to be a little quieter in conversations if being quieter suits your personality. But spend that time studying what other people do to connect. Observing them helps you figure out what's appropriate for the setting and lets you prepare to be more actively engaged at the next gathering.

BUSINESS CARDS

If appropriate for the situation, exchange business cards as part of your introductions. But don't hand out cards to everyone you see, whether you speak with them or not. That kind of canvassing feels more like marketing than networking. If someone tells you something you want to follow up on, ask whether it's okay to jot a note on the back of their card. That way, you're seen as taking an active interest in remembering what they told you, and you're not seen as carelessly defacing their card.

3.5.2 *On LinkedIn*

LinkedIn is probably the top social media platform for businesspeople, which is why I'm emphasizing it here over Facebook and Twitter, but these tips apply pretty well to any social media platform:

- I try not to accept connections from anyone I'm not willing to engage in a direct chat. That's kind of like saying, "Hey, you're welcome to step into my office, but I'm going to ignore you the whole time," and that's not the impression I want to offer as part of my professional brand. If someone turns out to be a spammer, I can simply reply, "This isn't how I use LinkedIn, and while I appreciate your time, I'd like to stop this." Then I can unconnect or take other appropriate actions. But in accepting a connection, I try to assume positive intent and always reply to direct messages.
- I use both Twitter and LinkedIn to promote the things I'm doing and seeing in my communities: new books, blog posts, podcast episodes, conferences I find interesting, code projects I've been using, and so on. This approach lets me promote not only my own work, but also the work of others, because I want amplifying others to be part of my professional brand. I tend to shy away from personal stuff, as I've mentioned before, and to keep my posts focused on my business.
- When I reach out to someone else, unless they're an existing colleague or co-worker, I include a small note letting them know why I sent the connection request. The note could be as simple as "I've been reading your blog, and I'd really like to maintain a professional connection with you," but it helps communicate my intentions.
- My rule for commenting on other people's posts: if you don't have something nice to say, say nothing unless they've explicitly asked you for complete feedback. There are more than enough people willing to tear others down, and I've felt no need to help out in that regard.

Remember that everything you do on the public internet becomes part of your professional brand. You can delete stuff later, but it's safe to assume that someone else has already picked it up and archived it. Whatever happens online stays online, forever.

3.6 **Becoming a confident networker**

When you've found your networking group or event and are in a small group, either online or in person, what do you do or say? Making conversation with strangers does not come naturally to everyone. Here are some tips for being a more confident and effective networker:

- *Make sure that you understand what your brand is and that you represent that brand during every one of your interactions.* If people see you in one context as being a helpful, engaged member of a technology community and in another context as being a gossip who speaks poorly of other people, they're not going to form a good impression of you, and they won't want to be part of your network.
- *Develop a concise introduction for yourself.* Be able to accurately describe who you are and what you do in a couple of sentences. If you're actively seeking new work, make sure that you can succinctly describe what you're looking for.
- *Get comfortable walking up to strangers—say, at a conference or user group meeting—and introducing yourself.* Being confident enough to say something like “Hi! I'm Don, and I do a lot of work with PowerShell. What do you do?” is a simple way to start a conversation.
- *If you are an introvert, you may have to force yourself to do in-person networking.* Try to show up at networking opportunities with a good amount of energy, and expect to feel drained afterward. The effort is worth it.
- *Ask questions.* Humans tend to engage more rapidly and deeply with people who aren't talking just about themselves, but who show interest in other people and organizations. Networking is a great way to gain information by asking questions like these:
 - “What's it like to work at your organization?”
 - “What are some of the qualifications you had to meet to earn your current role?”
 - “What sorts of roles did you hold before this one?”
 - “Are you aiming your career in a particular direction right now?”
- *When someone introduces themselves to you, repeat their name* (“Nice to meet you, Jason!”). Focus on that repetition; don't let it become a habitual thing you do. The idea is to make your brain pay attention to the person's name and face, and start making a connection. Then use that person's name a few times in immediate conversation (“Jason, tell me a bit about what you do”) to cement the name in your brain even more. This tip helps you remember people's names—something that always makes a good impression.

- *Concern yourself more with collecting business cards than distributing them.* After you speak with someone, make a few notes on the back of the card about what you talked about. After the event, write a short personal email to each person you met, thank them for the conversation, and let them know that you look forward to meeting them again.
- *Offer to help.* The best way to make someone feel positive about you is to help them. If they have a technical problem, offer to meet online later to work on it with them, or offer to pull out a laptop right then and take a look. If they're job-hunting, offer to go through job postings together. Find ways to help others, and they'll almost always be willing to help you later.
- *Finally—and this is the big thing—make sure that you have plans to stay in touch.* If you meet new people in your specific field, why not try to put together a monthly user group call, on which everyone can discuss what they're up to, share problems, and offer solutions? Create opportunities to reconnect in a meaningful and helpful way. Join open source projects that your contacts contribute to, and start making your own contributions; even code reviews and documentation proof-reads are helpful.

3.7 Action items

For this chapter, I'd like you to work on a networking plan:

- Start by setting some monthly networking goals for yourself, such as “Get to know three new people each month” or “Attend an in-person event and develop five new high-quality connections.” Emphasize quality over quantity! Don’t forget to include goals for maintaining and connecting with your existing network.
- Evaluate any networking activities you already engage in. How did you start them, and how are they performing for you in relation to your goals? How might you continue them in the same way or alter what you’re doing?
- Identify some new networking activities that you might try, with an eye toward meeting your networking goals. You’ll need to give most new activities three to six months before you can really evaluate their effectiveness.

Be part of a technology community

I will almost guarantee that you've been the beneficiary of a technology community. Whether it was someone answering a question you posted on [StackOverflow.com](#), a blog post that taught you a new technique, or a YouTube video that finally cleared up your confusion about a technology, communities have been there for all of us. But to have a truly successful technology career, you've got to do more than just consume what a community has to offer; you have to be part of it.

4.1 *The value of community to your career*

When I use the phrase *technology community*, I'm referring to a group of people—often spread across the globe—who come together in a set of shared spaces, both physical and online. They share information, develop professional relationships, and support each other. Many technology communities are difficult to see, because they don't create a dedicated "About Us" website for themselves. Instead, they exist somewhat ephemerally, their activities spread across numerous online and in-person forums.

Consider the community I've been a part of for much of my career: PowerShell. This Microsoft-created technology has developed a huge following over the years. You can see evidence of this community on Twitter (look for #PowerShell), on websites like [PowerShell.org](#) and [PowerShellMagazine.com](#), in the blogs of hundreds of individual contributors, at conferences like PowerShell Summit, and in the many in-person and virtual user groups that meet every month. Although this community has no central hub, if you start looking across its venues you'll start to notice a lot of familiar faces. Names like Adam Driscoll, Missy Januszko, Chrissy LeMaire, James Petty, Mike Kanakos, Jeffery Hicks, Jan Egil Ring, Tobias Weltner, Jason Helmick,

and many more pop up in many of the same places. They're often the ones posting news about the technology, announcing user-group meetings, answering questions in Q&A forums, or speaking at conferences. A huge, almost invisible crowd gathers around those commonly seen names. Hop into the Q&A forums on [PowerShell.org](#), and you'll see names and handles like Matt Bloomfield, gorkkit, kyprasoon, Doug Maurer, and more. You'll see links to blog posts by people like Mike Robbins, Kevin Marquette, and Jonathan Medd.

Very few of these people would consider themselves to be community leaders, but they are. **They're the ones offering help, taking the time to organize user group meetings, volunteering to run conferences, and filming videos that you can watch for free on YouTube.**

Almost every one of the names in the PowerShell community started in the same place: asking a question on a Q&A forum, reading their first blog post, or watching their first video. Most of them came to a community that already existed and began participating in it—often as consumers of information, the people benefitting from what the community offered. Over time, these big names decided to try to give back. They started answering questions in addition to asking them. They started posting their own blog articles. They started volunteering to run conferences, user groups, and online meetups. They were inspired by the contributions they had benefited from, and they wanted to give *back*.

And in giving back, they've almost all given a considerable boost to their own careers. Some now work for the PowerShell team at Microsoft, having caught the company's eye through their community contributions. Others have gotten promotions or new jobs because they found employers who valued team members who were so willing and able to help others. **Most of all, each of the names in this community, simply by choosing to participate and help others, has built an incredible network of peers and colleagues. Those networks support and enhance their careers in ways that are difficult to describe. I imagine that most of them, if they needed a job, wouldn't have to do much more than send out a tweet, and they'd have recommendations, leads, and even offers within days.**

Community vs. network

The ideas and purposes of *community* and *network* overlap a lot, and it might be useful to think about how they differ and what they're each for.

A *community* is a group in which you participate. You consume information from the community, and ideally, you offer new information back to the community. That information, as I've already discussed, can include code contributions, answering questions, teaching, and other activities. You might not know everyone in the community, because some of them might be passively consuming information and still working up to being contributors themselves. Communities are usually visible to the public, something anyone can jump into and participate in.

(continued)

A network is a smaller group that you cultivate for yourself. It's people you know personally and with whom you likely share many professional goals. They're the people who rely on each other at a more personal level, perhaps for things like job recommendations or career advice. Your network might be hard for someone not in it to perceive; professional networks are less visible than public communities.

These technology communities exist everywhere, all the time, for every technology you can possibly imagine. They exist for things other than technologies, too, such as for shared concerns like diversity, career growth, and other adjacent topics. You're likely already interacting with them, not realizing that the person who posted an answer to your Q&A website question is part of something bigger—and that you can be part of it too.

Let's be honest about our industry: vendors and open source projects are pumping out new technologies so fast that hardly anybody can keep up. Documentation is getting slimmer and slimmer. The only way any of us can get through the day, most days, is with the help of other schmucks just like us. We run Google searches, we post questions, we search YouTube, and we hope that conference we couldn't make it to recorded their sessions so we can access that information on demand. Without community—without *each other*—we'd all be up a creek without a paddle most days.

You can simply consume the output of those communities. You can be a passive participant, reading others' answers, watching their videos, or lurking in a virtual user group meetup. But in nature, there's a word for an organism that consumes without giving back: *parasite*. It's a word with a lot of negative connotations, obviously, and it's unpleasant to think of oneself as such an organism. Fortunately, when it comes to technology communities, you've got an easy way to do something better.

You can contribute.

4.2

Yes, you're worthy of contributing

I have this “contributing” conversation all the time, and over the years, I’ve probably spoken to tens of thousands of technology professionals about it. The almost-knee-jerk reactions tend to fall within two main categories:

- “Oh, I don’t really have anything I could contribute. I’m just getting started.”
- “Oh, I really don’t have the spare time to contribute.”

Both of those attitudes, if you happen to share them, are insulting to you. First, you’re probably not considering the sheer breadth of valuable contributions that exist. Perhaps you benefit from reading blogs but can’t picture yourself as a blogger. Maybe you get a ton of value from a community-run conference but can’t imagine why, or how, you’d want to run a conference of your own. Broaden your scope: just because you aren’t able to contribute in a way that has been valuable to you doesn’t mean you can’t contribute in ways that are valuable to others.

Second, you're likely suffering at least a little from Imposter Syndrome, that pernicious feeling that you're the least-capable person in the room and that everyone else is going to figure it out if you call attention to yourself. When it comes to community, though, everything is good enough. Perhaps you've just figured out how to do something with a particular technology; why not blog about it? It doesn't matter if 100 other people have already done so; **your perspective is unique and valuable**, and may help someone else in a similar situation. Your contributions don't have to be impressive to the people you admire; they simply have to be valuable to someone. And there's always someone with a bit less experience than you who could benefit from your help.

Third, you do have time to contribute. If you had the time to consume other people's contributions, you need to make time to pay it forward and help out. Nearly 100% of all technology communities exist on the backs of volunteers, who are using their free time to help you. You owe it to them to turn around and help someone else. If you really consider the breadth of opportunities for contributing, I bet you'll find something you can do.

So please accept these facts: you are worthy of contributing, your contributions are needed and valuable, and you can find the time to give back to a world that's probably already given you a good bit of help.

4.3 **Ways to contribute and participate**

I don't normally love to write long lists of bullets—and I suspect the editors of this book will be giving me the side-eye for this one as well. But in this case, I want to convey the enormous range of ways in which you can contribute value to the technology communities that you inherently belong to. Some are obvious ways to contribute, but I also include the less-obvious ways, which are every bit as needed and valuable.

Please take the time to scan this entire list. The idea is to help you broaden your mind on what *community* and *contribution* mean:

- **Blogging**—Even short articles about how you solved a problem are tremendously useful. Don't have your own blog? Blogs are free and easy to start on many platforms, but you might also look for a community website that lets its members blog there, putting you more closely in touch with the people your blog might help.
- **Videos**—Short demos of how to perform particular tasks are always useful. YouTube and Vimeo are popular video platforms, and you can use your social media reach to draw attention to your efforts.
- **User groups**—Help organize them, find guest speakers, and spread the word.
- **Q&A**—Find one or two sites that cater to your technologies, and jump in. Even if you're not the first to answer a question, you can always add more perspectives, options, and explanations to others' answers.
- **Translation**—Blogs, videos, and open source e-books can all benefit people who don't speak or read the original language. If you speak a second language,

contact the author(s) and ask whether you can help localize their work to broaden their audience.

- **Open source projects**—Contributing code is the obvious way to contribute, but it's hardly the only way. Most projects are in dire need of
 - Documentation updates and fixes
 - Unit tests
 - Localization
 - Issue moderation
 - Pull-request reviews
- **Podcasts**—Not ready to run your own? See whether one you really like needs help, such as with scheduling guests or writing show notes.
- **Conferences**—Maybe you don't need to start one, but you can volunteer to help an existing one. Conferences usually need gofers, registration workers, panel moderators, and many other roles.
- **Open source books**—You can write books if you want to, but you can also contribute by offering to proofread, translate, fact-check, and handle other tasks.
- **Spreading the word**—One of the most popular features of [PowerShell.org](#) is a weekly “In Case You Missed It (ICYMI)” roundup, in which the authors pull together a list of notable posts from that week. They help amplify others’ voices and connect community members with new people and new contributions. You can use your social media reach (even if it’s not huge), your own blog, and any groups in which you participate to help spread the word.

I was delivering a closing Q&A and suggestion panel at PowerShell Summit one year. We’d run an “Iron Scripter” coding contest earlier in the week, and during my panel, one of the attendees suggested making it an annual event: “We could organize regional events and get more people involved. Perhaps the winners at the regional events could get a discounted admission, or even a sponsored trip, to the next Summit, to compete in the finals.”

“That sounds wonderful,” I said enthusiastically. “You’re in charge.”

Everyone laughed a bit, and some people clapped, but my point remains valid: there are lots of things that communities can do together to create professional connections, improve their craft, and help each other. But unless someone does those things, they won’t happen. Don’t look around and wish someone else would do something to benefit you. Get up and do it yourself, to benefit yourself and others.

4.4 **Etiquette for participating**

Like networking, participating in technology communities comes with a certain set of global expectations that you should strive to follow. These expectations support a positive professional brand and help make you an even more valued member of your communities.

4.4.1 On Q&A websites

Q&A websites are a special kind of social media. If you've used them, I'm sure you've seen examples of both incredibly positive and incredibly negative behavior. I started [PowerShell.org](#) in large part because I'd seen so many other Q&A sites exhibiting overwhelmingly negative behavior, and I wanted to offer a better experience for people working with that topic. Be mindful that your behavior on Q&A sites is very much a part of your professional brand, and act in a brand-positive way:

- ***Don't be the person who only asks questions and never offers answers.*** Even if someone else has correctly answered a question, you can add expanded explanations, alternate approaches, and more so that you're contributing.
- ***Don't reply to a question with a dismissive or insulting phrase*** like "Did you try to Google that?" Assume positive intent, and assume that the person asking the question did indeed try that and didn't get the answer they needed. If you can't or won't offer an answer, say nothing.
- ***Don't engage in unprofessional negative behavior.*** Act like you're in an office, physically present with the people you're typing messages to. "That's a stupid answer" is unacceptable in real life and online. "I disagree with certain parts of that answer, and let me try to explain why" is a much more professional approach for expressing a different opinion.
- ***Make sure your answers are as complete as possible.*** Ask follow-up questions of the original poster, if necessary. Provide links to documentation or other resources, if appropriate. Being a solution is a pretty worthy goal for any professional brand.

4.4.2 In open source projects

Participating in open source projects is a wonderful way to engage with community, exhibit a positive professional brand, and make a real difference. Just keep in mind the rules of the road:

- **Most projects will have documentation on how people can participate. Follow the rules to the letter.**
- **Consider all the tips for Q&A websites (section 4.4.1) as you post issues, answer questions, and perform other activities within the project.**
- **When posting issues, make sure you're providing all the information that the project's maintainers request, including reproduction steps, screen shots, example code, or whatever else they've indicated would be useful. Take care to research past issues before posting to understand whether your situation has already come up and been addressed.**
- **When contributing code, make sure you've taken the time to understand and follow any existing naming conventions, coding patterns, or other practices that the project follows—both documented and inferred from the existing code. Make maintainers' jobs easy by thoroughly testing your code, writing unit tests as appropriate, and being a good coding citizen.**

4.5 Action items

For this chapter, I'd like you to look around the technology communities you already interact with. They might be Q&A sites, they might be blogs or other websites, and they might even be in-person gatherings like user groups or conferences. With all those communities in mind, consider these questions:

- Who are the leaders of those communities? Are there any high-visibility individuals whose contributions or participation really stand out to you? What do they do that you might consider doing yourself?
- **What value do people get from those communities? Education? Assistance? Networking opportunities?**
- **Where could you begin contributing immediately? Don't throw up your hands and say, "Everyone is already doing something, I can't possibly contribute anything more!" You know you can.**
- **Create a contribution schedule for yourself. Commit to being part of your technology community on a regular basis, not necessarily daily, but if you're really thinking about your career, at least a few times a month. Make a schedule, put it in your calendar, and commit to it.**

Keep your tech skills fresh and relevant

We all know that the tech world is constantly moving and changing, and you probably realize that the skills within your career need to keep up with the changes. That means not only updating your skills as the technologies you work with evolve, but also coming up to speed on new, relevant technologies that affect your career.

5.1 Fresh vs. relevant

I try hard to think of my skills in terms of my career, not just my current job. That is, while my employer obviously requires certain skills that I need to keep fresh or up to date, my career may require a different, and often larger, set of skills to remain relevant in the marketplace.

One of my first tech jobs was as a systems operator for a company's IBM AS/400 midrange computer (now known as the IBM iSeries). I had certain job skills that I needed to continually develop and keep fresh: the OS/400 Command Language, the various ongoing changes IBM would make to the computer's operating system, and the occasional changes to the hardware itself. My employer was responsible for helping me keep those skills fresh by sending me to classes and buying me books.

In the broader realm of people who operate computers for a living, however, AS/400 was a dead end. The field wasn't growing; people weren't buying new AS/400s. In fact, their almost-total reliance on the AS/400 would eventually prove to be a competitive disadvantage for the company, as new competitors came in with newer technologies that let them be more competitive for a lot less money. I probably could have eked out a 40-year career at my employer, doing nothing but operating that old AS/400, but I would have been entirely dependent on that company for my income, because there are relatively few job opportunities in the AS/400

space. In reality, my employer was eventually bought by one of its competitors, and the AS/400 was eventually decommissioned. That would have been terrible for me! With only that one skill, which had little relevance in the marketplace, I would have been hard-pressed to quickly learn new, more-relevant skills so I could have gotten a new job.

So the first axis of career skills is keeping your existing, employer-required skills fresh. The second axis is making sure to keep your career skill set relevant. **Your career skill set will often need to be larger than the skill set your employer requires and may require a significant personal investment of time and money to maintain.** Although your employer is welcome to stick with a given set of technologies, even if it hurts the company's competitive edge at some point, you need to make sure that your career can stay relevant in a competitive, ever-changing technology market.

You can express this idea as a chart (figure 5.1), which can be a great way to help you decide where to focus your learning efforts.

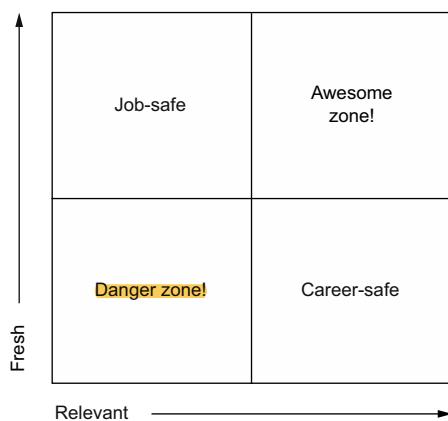


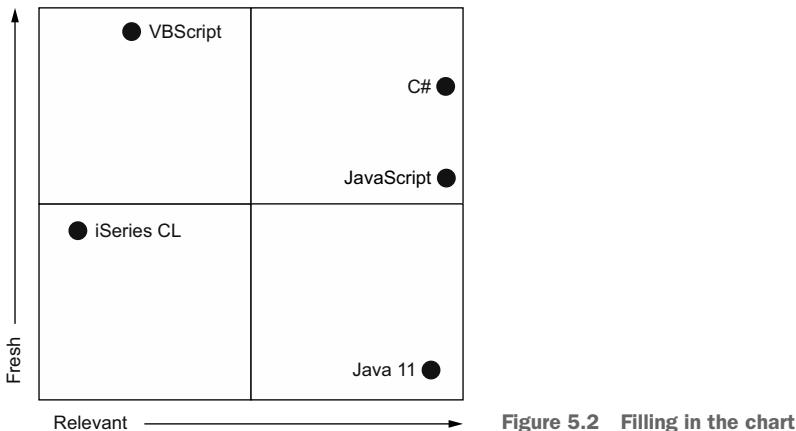
Figure 5.1 Charting your career skills

I'll categorize my skills in these four zones:

- **Job-safe skills** are ones that are fresh, meaning that I can hold down my current job, but they may not be fully relevant in the rest of the industry. These are skills I might have on maintenance mode, meaning that I keep up enough to maintain my job but don't otherwise invest in them.
- **Career-safe skills** are ones that are super-relevant in the industry, but I might not be fresh enough in them to hold down a job, and they're perhaps not needed by my current job. These are skills I won't actively focus on, although **I try to make sure I know how to come up to speed quickly should I need to.**
- **Danger-zone skills** are neither fresh enough to maintain my job nor relevant in the industry. I need to work on these skills, but probably only as much as my job needs. I don't need to be any more proficient than my job requires.

- **Awesome-zone skills** are relevant in the industry, and they're ones where my knowledge is fresh and up to date. These skills are the ideal ones to have because I can use them in my current job and in other jobs if needed.

Here's how I might fill in that chart (figure 5.2).



Here's what that means to my learning plans:

- I'm great at VBScript, but it's not relevant anymore, so I'm not going to spend much time on it **learningwise**.
- iSeries CL isn't relevant anymore either, and I'm not nearly as up to speed on it. If my current job requires me to learn more, I will; otherwise, I won't focus on it.
- C# and JavaScript are super-relevant, and my skills are reasonably up to speed. These skills are strengths that might help me land a new job. I want to keep my hand in on these technologies, perhaps by working on community projects in my spare time, even if my current job doesn't require them.
- Java 11 is a highly relevant skill that I know little about. That might be okay. Maybe Java 11 is a direction I don't truly want to pursue, but it's might enhance my career. I need to make some thoughtful decisions.

Keeping my employer-mandated skills fresh has always felt straightforward. A variety of training media exists in the world (and I'll get into some of them later in this chapter), and I can use whichever ones my employer and I agree will get the job done. Typically, my employer will pay to keep those skills fresh, making the process easy.

I've had a harder time keeping my career skills relevant. It can be tough to decide what *relevant* means! That's what I want to touch on next.

5.2 Deciding what's relevant

What skills are required to keep your career—to keep you—relevant in the broader global marketplace? Because you own your career, and because your career is intended only to take you to your success, only you can decide what *relevant* means. **The process begins by understanding what problem you solve for employers.** Do you maintain networks? Write proprietary desktop applications? Create web applications? Keep servers up and running? Whatever your broad solution area is, that's where you need to focus. Few of us can be true jacks of all trades these days, so you'll want to keep yourself centered on the broad problem space you solve for.

YOU CAN ABSOLUTELY SWITCH BROAD SPACES I don't mean to imply that, for example, a network engineer can't ever become a software developer. You absolutely can! But you're changing fields at that point and in many ways changing your career. It's a bigger undertaking and not a process I'll touch on in this chapter.

When you identify the broad space you're in, you need to start looking at the market trends in that space. **What technologies are employers hiring for?**

One way to answer this question is to rely on your professional network. Chatting with colleagues in other companies can certainly help give you a broader view of what's going on in the industry.

I also gain perspective by browsing job listings from all over the country and even all over the world. I use a variety of job posting websites—as many as possible, so that I'm getting the broadest perspective possible. But I try not to be overly influenced by shiny new trends. Just because Google released yet another JavaScript programming framework, for example, does not mean I feel compelled to sit down and learn it. Instead, I want to see what kind of market penetration that new technology gets. When I see lots of people starting to ask for it in job postings, I'll know I need to get up to speed.

Instead of the new and shiny, I look for technologies that appear again and again in job listings. **If lots of people are hiring for skill in a technology, by definition, that technology is relevant in the current market.** I'll be honest: my approach can result in **my being a smidge behind the curve,** in that I'm waiting for employers to adopt a technology rather than predicting what I think they'll adopt. But my approach is the only way I'm able to limit all the things I have to learn to some reasonable set that is likely to enhance my career.

5.2.1 Proficient, not expert

I think it's important to recognize that you do not need to become an expert in every skill employers in your field are hiring for. First, it's entirely valid to choose a reasonable subset of those skills. Second, you need only basic familiarity with the skills you choose, along with a strong set of learning muscles.

I like to visualize each of my tech skills on a breadth versus depth continuum to determine what I know and what I need to know. This approach aligns with the knowledge pyramid created by Mark Richards and Neal Ford, which you can see at <http://mng.bz/eMBJ>. Their pyramid, now familiar to many in the tech field, looks something like figure 5.3.

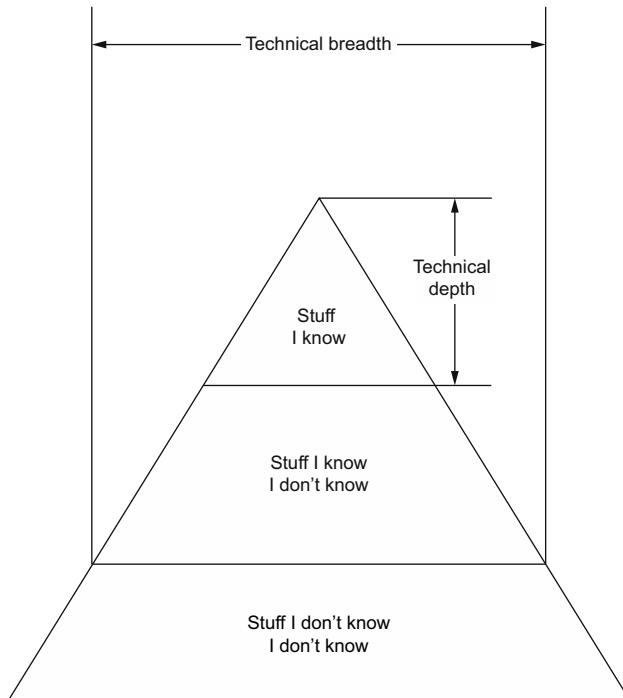


Figure 5.3 The knowledge pyramid

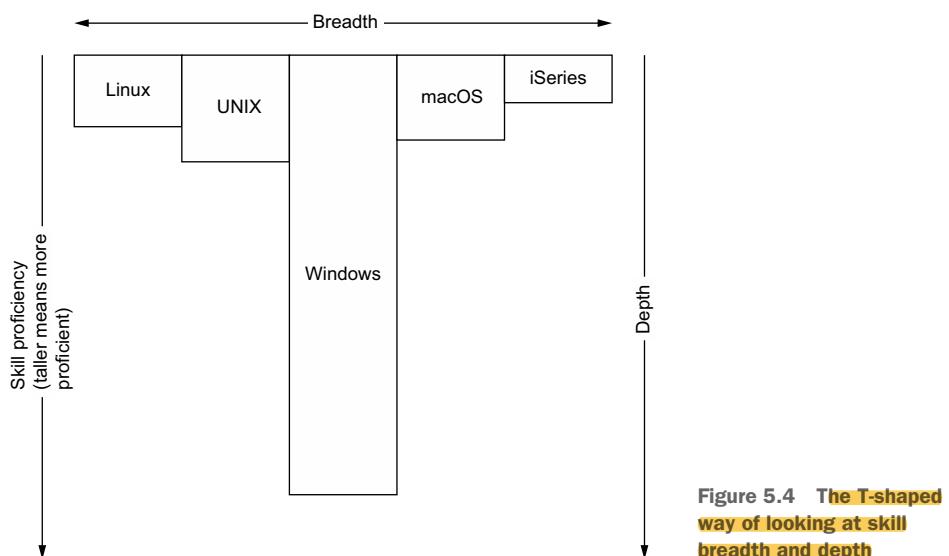
The idea is that in any given technology, there will be all the stuff you know how to do—the skills you have. That's your technical depth for the subject. For some subjects, technical depth will represent a large chunk of the pyramid; for others, the chunk may be more like a *sliver*. In addition, there's generally a bigger chunk of stuff about technology that you know exists but that you're not confident doing; that stuff represents your technical breadth, or the areas where you could become proficient, given a little time. Then there is the chunk of things that you don't even know you don't know—the stuff about the technology that's beyond your *reckoning*. For technologies in which you're a strong practitioner, this chunk might be fairly small. My goal here is to

- 1 Identify the skills in which I need strong technical depth to get or retain a job.
- 2 Identify the portions of the skill that I need to know to get or retain a job.

- 3 Learn the skills and skill components that I identified in steps 1 and 2. As I learn them, I'll pick up a lot of the second-layer "stuff I know I don't know" knowledge and skills, and I can make those targets for future learning to make myself even more proficient.

Step 2 is the hard one. What, specifically, do I need to know about JavaScript or PowerShell or microservices to do a particular job? One way to approach this question is to use industry benchmarks. Certifications, nanodegrees, and other kinds of external measurements can help define that top layer for you, helping you figure out where to focus. Some portions of the tech industry (such as IT operations) have lots of certifications; others (such as development) have fewer, but if you poke around, you'll tend to find plenty of certification-like benchmarks for anything. Some training companies offer assessments, such as Pluralsight's Skill IQ; others offer nanodegrees, and so on.

Another way to look at proficiency is to examine your level of knowledge in the "stuff I know" chunk of the pyramid. All of us have skills in which we're strong and ones in which we're so-so. This way of assessing our proficiency is sometimes called a T-shaped skill set, which you can see in figure 5.4.



The T-shaped chart is just another way of looking at the pyramid, except that you assess several skills instead of one. The skills for which the list of "stuff I know" is big are where my skills are deep (as in Windows), and they appear as taller bars on the T-chart. Considering all the skills about which I know at least a little, you've got my breadth, represented by all the bars on the chart. I can increase the depth of any of my shallow skills (Linux and iSeries, for example) if I need to. That's where the scatter

graph and its four quadrants come in handy: I want to focus my learning efforts on the skills on which I'm currently shallow and those that are highly relevant in the marketplace.

5.2.2 **Where to focus**

When you've thought about the skills in which you're deep and the ones in which you're not, and you've looked at which skills are relevant in the marketplace or needed by your job, you've got the beginnings of where to focus your learning time.

You can also focus some or all of your learning time on topics that are of general relevance in the industry, depending on your job role. For that approach, **start by choosing a reasonable subset of skills in which you want to stay relevant**. That can be tough, because any field of technology has a set of competing choices. Here are some examples from a variety of fields:

- **Software developer**—Languages include Java, C#, JavaScript, Python, PHP, and hundreds more.
- **Systems administrator**—Operating systems include Windows, Linux, UNIX, and older mainframe and midrange operating systems.
- **Network engineer**—Brands include Cisco, Juniper, Aruba, and many more.
- **Database administrator**—Choices include Microsoft SQL Server, Oracle, MySQL, PostgreSQL, and dozens more.

All these technologies are important and popular, and you'll find many employers hiring for any of them. So how do you choose? Do you have to become an expert in all of them?

To identify my reasonable subset of skills, I start by looking at market share to find out the popularity of a language, system, or tool. This task is often as simple as searching online for something like “networking vendor market share” or “cisco market share” and then focusing my research on one or two of the top competitors in the market. That approach serves two purposes:

- The top competitors are the ones where I'll most likely be able to get and keep a job.
- If I have strong familiarity with the top competitors, the lower-lever ones won't be hard to learn if I need to.

When I've identified my target technologies, I remind myself that I don't need to become a world expert in them. I just need basic, core familiarity and confidence in my learning muscles' ability to help me learn more quickly.

For me, *basic familiarity* usually means whatever I could get from 40 to 80 hours of instruction in the technology. That might not be enough to get a job, but it's enough that I could direct my future learning in the right direction and quickly gain more knowledge so that I could get a job. Presuming that I scheduled two or three hours a night for three nights a week, I could get in 60 hours of learning in a couple of

months. To me, that is a reasonable investment of time for a reasonable return of skills and knowledge.

There's an upside to aiming only for basic familiarity: the basics of any technology tend to change most slowly. That means my learning investment in getting to that point will last longer, making it easier for me to keep that basic familiarity fresh, and giving me time to explore other topics to the same level of knowledge. My reliance on getting only to basic familiarity depends, as I've indicated, on my confidence in my learning muscles.

5.3 **Building strong learning muscles**

Everyone is familiar with the basic process of building muscle: you have to work those muscles every day, all the time. People sometimes talk about being lifelong learners, but I prefer to think of myself as a daily learner. **Your brain's ability to learn, and to learn quickly, is something you can train and develop, and the way to do that is the same way you build a stronger heart: moderate, daily exercise.**

I schedule time every day to learn, often during my lunch hour, because I enjoy reading and that hour is a great time for me to read—and a great time to take a break from work and think about something else. But sometimes, that learning time is in the car on a long drive, when I can listen to a podcast or the audio track of a training video. I rarely schedule my learning time at night, because I know that I don't retain information well past dinnertime.

What do I learn? Surprisingly, not always technology! Sometimes I read a news article and go down a rabbit hole. I enjoy reading about US Supreme Court cases, for example. I'm not a lawyer, and the articles often reference legal principles and cases that I'm not familiar with, so I learn about them. I'll Google legal principles and run across briefings and other short documents. I'll Google earlier cases as well and read articles about them. Before long, I've learned something and spent only an hour or so doing it.

If you're trying to build a stronger heart muscle, the muscle itself doesn't care if you're running, walking, hitting a punching bag, swimming, or engaging in some other activity; it knows only that it's being exercised, and it responds by slowly becoming stronger. It's the same with learning: your brain doesn't necessarily care what you're learning, only that you're exercising the mechanisms that it uses to learn. So I don't always try to focus on a new technology skill. I do that all day at work, and some variety is nice now and then! But I'm learning something every single day.

That exercise has given me the confidence that I can learn new technology skills or improve a given technology skill rapidly when I need to. I once taught myself enough Python in a few hours to solve a significant problem at work, and I'd never used Python at work before. I knew a little bit about it, because it's relevant to my field, and relied on my learning muscles to get me the rest of the skill I needed. I've learned JavaScript, C#, PHP, Linux, PowerShell, Windows, SQL Server, MySQL, Cosmos DB, and dozens of other technologies in much the same way.

My strong learning muscle means that I'm confident with having only basic familiarity with the key, market-leading technologies in my field, because I know I can learn more, and do so quickly, when I need to.

5.4 Learning media

We're all familiar with the "classic" learning media in the world of technology: instructor-led classes, self-paced videos, and books. These media are effective, and you should definitely use them when they're appropriate. But I don't want you to discount all the other fantastic ways to learn, especially when you're learning to keep up versus building a new skill from scratch.

Vendor documentation is the first on my list of alternative learning resources. I know the quality varies drastically across the industry, but being able to learn from vendor docs is an incredibly important skill in our industry. If you can go to that source material, digest it, and construct new knowledge for yourself from it, you'll be unstoppable.

I also use the internet, often for broad introductions to new technologies, features, or approaches. When I'm trying to develop a basic understanding of what something does and how it does it, I'll start in a search engine. I'll open tons of blog articles, Wikipedia articles, YouTube videos, and other bits of content. I'll skim those articles rather than reading them closely, with the goal of finding ones that explain the topic at the level that works for me and what I already know, as well as how deeply I want to understand it.

Social media can be a learning resource too. One reason to build a strong audience on Twitter or other platforms is that you've got lots of people to ask for advice. If I'm trying to become familiar with a technology and can't find any reading material that's giving me what I need, I'll ask my friends on Twitter what they recommend. I'll have a dozen great recommendations within a day 99% of the time, which lets me focus my learning for the next few weeks.

My point is this: all learning is useful. You don't need to rely exclusively on the formal modes of education that we're all used to. If your learning goal is to keep up with a technology rather than master it, you can often invest less time, while still achieving the level of familiarity you want, by using more informal ways of learning.

I'll offer another tip: there's a pervasive belief that people tend to learn best from a particular set of learning media. In other words, you might feel that you learn best from reading, or from watching videos, or from attending an in-person class. That belief is a myth.¹ Human brains don't have any kind of built-in tendency that makes one medium more effective than another. Any brain can learn effectively from any medium, which is great news. You may need to exercise your learning muscle a little more to accommodate it to a particular medium, just as you might have to work up to running a marathon if you've never been an avid runner.

¹ "Belief in Learning Styles Myth May Be Detrimental," <http://mng.bz/pJZP>

5.5 **Assessing your relevancy**

One difficulty in keeping your skills fresh is knowing when you’re doing enough to keep your skills fresh enough or relevant enough. Keep in mind that **you don’t need to be a world-class expert in every possible topic to be relevant**.

I like to divide my skills into three broad categories, and I think this exercise is useful for making sure you’re not attempting to overinvest in your skills:

- **Skills I need for my job**—These skills are the ones I need to invest in most heavily, and I would expect my employer to co-invest with me, because these skills are enhancing the business. But—and this is a critical point—my employer needs to invest only to the level that the business requires. My employer and I might need to work together to determine when I have sufficient skill freshness, which might involve certification exams, internal or external skill assessments, and other tools. The exact tools you rely on together will differ greatly across technology disciplines, and you’ll need to do some research to discover what’s considered industry-standard in your part of the tech world.
- **Skills everyone else needs for their job**—These tech skills are the ones that the industry in general is using but that I’m not using in my own job. Because my employer doesn’t need these skills, I don’t expect the company to help me out here (although it’s great if it’s willing and able to do so). For me, these skills are the most critical ones on which to focus my resources, because these areas are where I risk falling behind. That is, if **everyone else is using these skills, and I’m not, the industry is probably moving in a direction that my current employer isn’t**. I risk losing my market relevance by ignoring industry directions. That said, I don’t necessarily have to master these skills. I simply need to get myself to the point of familiarity where I have some confidence that if I have to, I can rapidly reach basic proficiency, such as if I suddenly find myself in need of a new job. That’s a personal and subjective assessment, because you typically don’t need the level of proficiency that a certification would test for.
- **Skills that reflect general industry trends**—I look outside my own specific area of tech to see what everyone else is talking about—not necessarily what everyone else is doing, but what’s on everyone’s minds. As I write this volume, tech like quantum computing might fall into that area. For these skills, I try to achieve elevator conversancy. That is, if I can talk knowledgeably about the technology at a high level for about five minutes, I consider myself good enough. These are areas I tag to keep up on, usually by adding them to a reading list of topics that I revisit once a quarter.

5.6 **Tips for lifelong daily learning**

I’ve developed a few habits over the years that have helped me tremendously in keeping up with my career. I share them here, in the hope that at least a few of them will work for you as well:

- **Learning is a daily effort.** The way the human brain works means that you want to have it learning new things every day. That's how you keep the learning muscle strong, so that when you need to learn something, it'll come more easily to you.
- **Learning is not just tech.** I don't always focus my daily learning on tech. Sometimes, I just need a break from the computer stuff! So I might grab a random Wikipedia article, or go down a rabbit hole on something I read in the news. I'll find an interesting US Supreme Court ruling, for example, and read up on the legalities surrounding it. It interests me, and it expands my knowledge. You might follow another area of casual interest, reading up on real estate, art history, microbiology, or whatever sparks your imagination.
- **Schedule it.** If you don't make time for learning, it's easy for it to not happen. Identify the value that learning adds to your career, and commit to spending the time on that learning that it takes to achieve that value. Keep in mind that all your learning time won't happen at work: if it's benefiting your career, you're going to have to invest your free time.
- **Drag others into it.** If you have colleagues or co-workers who are in a similar place, form a small study group. Meet once a week, and teach each other something; spend the days leading up to that session learning what you'll be teaching. I've done this with small groups of four or five folks, and it's really effective. We'll divide a topic, and each person commits to a 10-minute teaching session. This approach lets each of us focus on a small area and adds structure to our learning efforts. Additionally, teaching our subtopics does wonders for reinforcing what we've learned in our own research.
- **Don't be afraid of random.** Although you should learn to do enough market analysis to decide what you should be learning, don't be afraid to roll the dice. I'll log in to Pluralsight and watch a random newly released course sometimes. It may not be in my field at all, but it's learning, and that's what's important on a daily basis.

5.7 Further reading

- *Mastery*, Robert Greene (Penguin Books, 2013)

5.8 Action items

For this chapter, I'd like you to start developing a personal framework for staying relevant. I hope you can start engaging in these activities now and make them part of your daily routine:

- Establish a daily learning schedule for yourself. Set aside time—an hour or two a day is sufficient—for learning. This time is an investment, and it may require you to give up something else, but this investment is worthwhile.
- Spend a few weeks' worth of your learning time analyzing your technology field, and come up with a list of the important, market-leading technologies you see

employers hiring for on various job boards. This list becomes your learning list. Don't let it be more than a half-dozen items at first: if it's longer, trim it by focusing on the most market-leading technologies you've found.

- Start spending your learning time achieving basic familiarity with the technologies on your learning list. Devote about 60 hours to each one. That's 360 hours for a six-item list, which will take you about six months at two hours a day. Don't focus on those technologies every day, however; give your learning muscles some variety by learning random topics as well. (The Random Article link on Wikipedia is a great way to find short new topics to explore.)

Show up as a professional

In addition to the razor-sharp technology skills you've built up to this point in your career, you need to bring something else to work every day: professionalism. We all work in a business environment, or in a businesslike environment. In this environment, certain behaviors help move your career forward; others hold you back.

6.1 Be your word

I've long followed three simple rules that I believe have contributed greatly to my success in business and in life. These rules may, in fact, be the Big Three that tell you everything you need to know to be successful.

First, a prediction: you might read these rules and think, "Well, duh." But they're not common-sense, "of course" rules. They're demonstrably not common sense because they're not that common. In fact, following these rules has enabled me, and past business partners of mine, to rise above the crowd simply because, as simple as these rules are, almost nobody follows them.

6.1.1 Never promise what you cannot deliver

It's a simple rule but hard to follow. Here's why: Most people don't want to be the bearer of bad news. If someone asks you to do something, and you say no, they will want to know why. So now you may feel that you are in a confrontation. Most people hate confrontation, so they'll say yes instead just to avoid one. People will say "yes" even knowing full well that they cannot deliver what they've promised.

So the first step toward being your word is getting better at saying no without creating a confrontation. I suggest adding a short, polite explanation to your answer to deter the other person from pursuing the request. Try saying, "No, because I'm

already completely committed. If I take on something else, I'll need to drop something I've already committed to do" or "No, I'm focused elsewhere right now, unless you can help me understand why I should shift my focus." Whatever the reason, don't say yes unless you will actually, for sure, definitely be able to deliver. Don't say yes thinking, "Well, I can probably cram this in someplace." Say yes only if you have a definite, workable, practical, realistic plan for making it happen.

This rule can be hard to follow at work, especially if your boss assigns you a task. You didn't really get a say in that, right? But you need to look at everything on your plate, go back to your boss, and be honest about what you can do. Let them know that taking on that task means you might have to drop another task ("I can get this extra coding done this week, like you asked, but it means I'll have to set aside the unit tests I was writing. Is that still okay?"). You might even show them the list and ask them which one you can postpone; this may help them empathize with your situation and see exactly how loaded your plate is. Unfortunately, in some organizations bosses try to load more on you and don't acknowledge that you can't do everything. If you find yourself in such an organization, maybe ask yourself why you're staying there.

6.1.2 Always deliver what you promise

This rule is the companion of the first rule. If you did promise something, no matter what was going through your head at the time, you have to deliver it. Period.

I've worked with colleagues who used the "My dog ate my PowerPoints that were on a USB drive" excuse. As you might guess, excuses like that diminished the amount of respect they received from their co-workers, managers, and bosses. People do not respect excuses, especially when those excuses are not credible. When people lose respect for you, your reputation—your brand—suffers, and the damage goes beyond you. Not delivering on a promise can have tangible, negative effects on colleagues, who might have to pick up the slack, and on achieving project goals and internal or external deadlines. If someone commits to do something for you, you want them to do it. Therefore, you have to do the same thing.

Delivering what you promise often depends on a key life skill: knowing what you can do. If I asked you to sit down and write a 10-page paper on a random topic, how long would it take you? Do you know? Do you know for sure? Many people simply don't pay attention to how long it takes them to accomplish things or make guesses instead of verifying. This leads them to not understanding what a commitment will actually involve, which leads them to overcommitting, which leads them to telling excuses and damaging their professional brand.

6.1.3 Be easy to work with

If you commit to doing something, make it as turnkey as possible for whomever you're doing it for. Nobody should have to bug you. Nobody should have to nag. Nobody should have to remind you. Nobody should get it half-baked from you, get into a fight with you about it, and make you redo it.

Easy, right? Sure. Except ask yourself how many times you've violated any one of these three rules. Be honest with yourself. Be brutally honest. If you can't admit your faults to yourself, you can't truly succeed at anything. So take a hard look in the mirror and ask whether these three rules are ones you truly live by in every aspect of your life.

Be your word. Nearly all my success has come from these three simple rules.

6.2 **Be detailed and precise**

Most successful people are pretty good at paying attention to details. But our brains often work against us when it comes to details, and losing the ability to focus on details can affect our ability to show up as professionals on the job.

Human brains have a good and bad little feature called *filtering*. This feature is deeply embedded; first and foremost, it helped ensure our primitive ancestors' survival in the wild. When you have the potential to become someone else's lunch, your brain has to decide quickly what in your immediate environment is important and what isn't. It needs to discard the unimportant things ("Ooo, pretty tree") and focus sharply on the important things ("Something is stalking me"). This can't be done as part of conscious thought, which would take too long; it has to happen automatically, constantly, and almost instantly.

Imagine yourself walking down a busy street in a city like New York or London. For the most part, your brain operates in macro mode, focusing on the big picture and the major threats: getting hit by a car, running into another person, tripping over a bad section of sidewalk. The details of the moment get filtered out: what other people are wearing, the sign on the side of the bus that just passed, the specials advertised in the shop window. But your brain has time for those details. After all, during that walk you're probably also listening to a podcast, texting on your phone, or engaging in some other activity. You're letting your brain filter remove the input that's considered unnecessary for the macro task of surviving the walk.

So recognize this: you can't shut off the filtering system any more than you can shut off your own heartbeat. The trick is to consciously focus the filtering on the details you want and need pay attention to.

I'm not suggesting that you stop listening to your podcast and instead pay attention to the fashion choices of everyone on your walk. I'm simply pointing out that your brain does this filtering whether you ask it to or not. Your subconscious is choosing what to focus on, and it does this all the time, in every task you perform every moment of the day. It does it during meetings, while you're working, and while you're engaging with your family at home. It's filtering out details that you'll never even realize were filtered because it's on autopilot.

But just as you can control your heartbeat, you can also exercise control over the filtering feature. Calm, deep breathing can help you slow and steady your heart rate, and physical training can help you moderate your heart rate and help your heart respond more appropriately to a given situation. The same is true of the filter. With some

active training, you can help your brain do a better job of deciding what's important and prevent the filter from trapping information that it should be letting through.

Here's why we need to learn to turn down our filtering instinct: we're not primitive hunter-gatherers focused on basic survival anymore. We have time to focus on the details. In fact, we must focus on the details to do our jobs. We must focus on which connection ports are we using, for example, or on the precise syntax in a line of code or commands, or on the time when a meeting will take place.

When I meet someone who asks how I like my iWatch, I immediately start to distrust them. "It's an Apple Watch, not an iWatch," I say to myself—and sometimes to them. Now, you may reasonably be thinking, "Don's kind of being a jerk, here. I mean, who cares?" You'd perhaps be right. But I care, and I'll tell you why: To me, this detail is indeed minor, so why not get it right? Exactly how much extra brainpower would be required to remember the correct product name? None at all. And given that no extra brainpower or smarts is required, all I can imagine is that this person isn't good at paying attention to details. Then I wonder what else they consistently get wrong. They don't have control of their filter, so what else about themselves can't they control?

This example is a good illustration of where your personal brand comes into play. If you create the public impression that you're poor at paying attention to details, that impression will become part of your brand, and that particular aspect of your brand won't help you in your professional life.

First (and second, and third) impressions are important, and there are myriad ways that you can put off other people. Many people have an instinctive distrust of someone who seems sloppy, for example, and that can extend to attention to detail. If you can't get the little things right, they may assume you're likely to not get the big things right either.

As you move through life, know that perhaps the most important bit of self-ownership you can have is over that brain filter of yours. Train it to obey you. Teach it not to filter out small details. Let those details flow over you, and actively decide what you need to pay attention to. Don't let some caveman decide what you're going to pay attention to.

How? Slow down. Our brain filters are designed to assess fast-moving situations quickly. They're designed to keep us alive during the hunt. Our modern world being so much more full of potential distractions, it's easy for our brain filters to shut them all out. Consider airports, one of the most frenetic and distraction-rich environments you can imagine. The overhead public address systems are always announcing something, you're rushing to your plane, people are everywhere and you're trying to dodge them, and your brain filters go into high gear. You start ignoring anything that isn't an immediate threat or obstacle. Admit it—you could be dashing to your flight, and someone could announce the recipe for curing cancer over the PA system, and you'd miss it. Slow down.

In my experience, when your body is no longer in a mad rush, your brain filters will relax. When your body doesn't feel like it's in a fight-or-flight situation, the survival

filters aren't as necessary, so they'll settle down. Your tunnel vision will expand to include more of the world, and your brain will become more observant. You're going to have to actively allow yourself to slow down. Leave 10 minutes earlier for that appointment. Get to the airport half an hour earlier, and allow for a longer connection. Don't rush. Over time, your brain filters will become less hair-triggered, and that will help you be more detail-oriented at work too.

Take a coding example. I've worked with junior programmers who had an error in their code and asked for my help. I looked at their code, and with my greater experience, I was able to spot the problem fairly quickly. I told them that there was indeed a problem and asked them to look more closely. But with me hovering over their shoulder, they were in an even bigger panic. Their filters kicked in, and they started rapidly scrolling through their code, looking for the error. They were in macro mode, trying to focus on the big survival picture, and their filters were trying to remove the details so that they could focus on the big picture—but that got in the way of their finding the problems. So I let that go on for a minute. Then I directed them to the region where the error was and asked them to walk me through the code. Almost invariably, they spotted the error when they come to it. They just needed to slow down and take in the details.

This example is very much like what brains did to keep cavemen alive, and it works against us in the modern, caveless world. Our brains are designed to look at a scene and remove the details: the blades of grass, the individual leaves on the trees, the sound of the breeze. Filtering out the details let our brains see the big threats, like the predator about to eat us.

But details are survival now, as we have ever-more-detailed and critical interactions with each other, with technology, and with our environments. So care about the details. A lack of attention to detail is what makes phishing email scams work. It's what makes nearly every scam possible, in fact. It's what causes car accidents. It's what makes you miss your train. It's what makes you miss that important meeting or overlook that crucial line in the output that shows where the problem lies. Nearly everything that can go wrong in your personal and professional lives can be traced to a lack of attention to detail. Train yourself to read the fine print of life all the time. If you find yourself skimming, stop, go back, and read it again more slowly. Focus on absorbing details instead of letting your brain do whatever it wants.

6.3 **Cut your losses when the time is right**

True professionals don't doggedly work the same problem forever. They realize that the outcome of whatever they're working on is most important, and they seek ways to get to that outcome as efficiently as possible. Sometimes, that actually means giving up.

Our brains don't like us to quit. Nobody likes to feel like a failure, and we often resist that feeling more powerfully than we would pursue a feeling of success. This way of thinking holds most people back in their lives. Rather than risk the feeling of failure, we content ourselves with a lack of success.

Knowing when to quit can apply to almost anything in a technology career. Maybe you’re working on a block of code that will not work. At what point do you erase it and start over fresh? Or maybe you’re trying to fix a server that keeps stubbornly rebooting for no obvious reason. When do you give up, reinstall the operating system, and start over? It can be frustrating to give up, but if doing so will get you to a solution faster, you’re actually winning.

You’ve probably heard Facebook’s catchphrase “Move fast and break things.” That’s a corollary to the idea of “Cut your losses.” It means that you try something, but you’re not going to keep trying it past the point of viability. Look at the major investments companies like Google have made and subsequently walked away from, such as Google Wave (whatever that was) and Google+ (more or less). You can emulate this approach. Try something, and if you can’t make it work, move on.

Yes, cutting and running is a kind of failure. Be okay with that. It’s not bad to fail. Failing is how we learn. It’s also not bad to fail and lose, if you’ve come to the point where winning is not going to be practical. Businesses do it all the time, and although bankruptcy can be scary, sometimes it’s the legitimate thing to do. Cut your losses, reorganize, and try again.

6.4 ***Let Blue Sky mode happen***

Don’t be a “no” person. Be a “how can we” person. Don’t kill ideas because the first iteration doesn’t seem workable. Be an engineer, not a roadblock. A negative vibe holds you back but nobody else.

Disney goes through a process for dreaming up new attractions for its amusement parks. This process is called Blue Sky, and it means that the clear blue sky is the limit. During Blue Sky discussions, employees are not allowed to say no to something or to start laying out how it won’t work. They dream. They say, “What if we . . .” and engage in pure speculation and invention. You don’t worry about the logistics in this phase. You don’t express your own likes or dislikes. You let the ideas come, and everyone in the discussion riffs to evolve ideas and express new ones.

Blue Sky mode rarely happens in the normal world. Sit in a conference room at almost any company and suggest a new product, and you’ll likely get immediate push-back—reasons why it can’t work, possible issues, roadblocks, and hurdles.

Don’t be that person. Don’t suppress other people and discourage them from contributing. Instead, let Blue Sky mode happen. Be the person who says, “Oh, it’d be hard, but maybe we could do [whatever] to make it work.” Be an engineer of solutions, not an engineer of roadblocks. Your idea for a solution might not be workable, but it’s still Blue Sky mode: let someone else take your nascent idea and pivot it to something that will work.

Normally, someone brings up an idea, and a bunch of other people crush it before it’s had a chance to live. Sometimes, they see the idea creating more work for them, and they don’t want to do it. Be honest with yourself if that’s how you feel. Tossing in reasons why not isn’t helpful, though, and it makes you the problem. It makes other

people not want to engage with you. It puts you on the outside. Don't worry too much about logistics when an idea is getting going. Instead, see where the idea goes. At the appropriate juncture, if an idea is getting solidified enough, start offering solutions ("You know, that would normally take more people than we usually can commit to it—but how about this. What if we [idea]? It might mean doing [alternative detail] instead—would that still hit the intent?").

Push other people to let Blue sky mode happen too. If you're the one bringing up the pie-in-the-sky idea, and someone else moves to crush it, take a moment to stop, examine that behavior, and advocate for the value of supporting new ideas rather than immediately destroying them. "Hey, we're looking for creative new ideas," you might say. "Let's take some time to work through them. Someone's first idea might not work, but if we all work on it a bit, maybe a fourth or fifth idea will be the right one. But we'll never know if we crush it out of the gate."

Bringing a solution (indeed, being the person who is a solution) in Blue Sky mode isn't useful only in work situations. It doesn't just win you supporters and friends. It's also a way to make your brain start to think about doing instead of not doing. Most people's brains are instinctively conservative; when we're confronted with something new, we often push back before we've even really thought about it. That's why most people are so resistant to change. The true innovators in our society don't say no. They say, "Hmm" and start thinking of a way to say yes. You have to be on the lookout for "I don't like new things" pushback that comes from the dark recesses of your brain. Recognize it for what it is, and set it aside. Deal with situations rationally, not emotionally or unconsciously.

Be a person who tries to find the yes. But if you can't, above all, don't be the person who tries to squash ideas. Let Blue Sky mode happen.

6.5 **Draw a yellow line**

Here's another Disney story. In this case, I love that Disney recognizes that familiarity breeds contempt.

What Disney sells, in its theme parks, is entertainment. It's not just about rides; it's also about an environment. Disney calls it a show, and shows, like all forms of fiction, require the willing suspension of disbelief. You know that the princess is really some college kid, but you choose to participate in the show and treat her like Cinderella or Minnie Mouse. A maxim of fiction is that for the audience to maintain their willing suspension of disbelief, you have to avoid chucking anything out-of-story at them. At Disney parks, you can't have an angry janitor who got dumped by her boyfriend the night before and who doesn't want to be at work scraping gum off the asphalt this afternoon.

All of us have worked in an environment where one or more of our co-workers got too comfortable and brought all their baggage or negative feelings to work. They marched in, plunked down at their desk, and made it clear that they weren't happy to be there.

Every time you set foot out of your home, you're engaging in a public performance, and everything you do affects everyone around you. That always-grumpy person isn't going to be first in line for a promotion because, honestly, everyone wishes they'd quit and go do something else. A surly middle-school teacher isn't going to be as effective as one who's in a better mood and who remembers why they're in that classroom in the first place. You need to present your best self every day, even if that's not how you really feel. Every time you drop the performance and let people see through the facade, you're breaking the story. You're hurting everyone else's ability to continue suspending disbelief. You're damaging your product.

That's why Disney theme parks have yellow lines. At any possible place where an employee can come from backstage and into the sight of paying customers, there's a line on the ground in yellow traffic paint. It's a hard visual reminder to Disney employees to leave their problems at that point. Maybe you had a bad night last night. Maybe you got dumped. Maybe your cat died. Maybe you're having trouble paying the rent. These experiences are bad, and it's fine to feel badly about them. Leave it all at the line, and it'll be there waiting for you when your shift is done, because when you cross the line, the performance is on. You smile, you briefly remember what you are being paid to do, you straighten up, and you go do it. "Past this line," you've been told, "you put on the performance, and you don't break character. When you come back to the line, you can go back to being whoever you really are."

Character. What an important word. In public, at work, we are all characters. We are playing a role. What does that role need to deliver? It doesn't need to be who we really are; we can be ourselves at home with friends and family. I'm a sarcastic, hard-to-get-along-with individual at home. I don't let that come through in my work character. Instead, my work character is the person my employer hired, and (I hope) a professional all of my colleagues—with their varied backgrounds and personalities—can work and get along with.

What most of us lack in life is that yellow line. After going to the same office and the same job with the same co-workers day after day after day after day, we get complacent. We lose our respect for the role we have at work. We forget who we are supposed to be at work, and we break character and drop our performance. That's when you damage Brand You. That's when your paying customer—your employer—gets to see behind the curtain. That's the precise moment when you take whatever seeds of success you've managed to plant, and you crush those tender little shoots under your foot. That's when your career declines into a mere job and you stop investing in your future. It's where you damage your chances of success.

In your mind, paint a yellow line. Perhaps it's inside the front door of your home. Perhaps it's outside the front door of your office. Wherever you paint it, mark it well. I'm serious: stand at that location and visualize a line, painted in thick yellow reflective traffic paint. Notice the scuff marks on it where people have walked over it time and time again. Notice the little nicks along one edge where the UPS guy wheels his hand truck every day. Make it so real in your mind that you can't not see it every time you

walk past that spot. Ask yourself how much longer it'll last before you need to touch it up. Make it real. Then respect it.

Every time you approach that line, think about what it means. Think about why you're at this job and what you hope to achieve from it. How is this job helping your career? How is this job helping you achieve your own success? How is this job enabling you to help others, either now or someday in the future? What is the point of it all? You don't need to be happy to be there—but you need to remember why you are there. Examine every bit of baggage you've got with you right then. Enumerate every negative thing plaguing you: cat died, got dumped, kid needs braces, car door got dinged. See the baggage, pause at the yellow line, and set down the baggage. Nobody will touch it. It'll all be waiting for you when you come back out, but it has no place on the other side of the line. Your performance is about to begin. Review your lines. Put a smile on your face. Raise the curtain, step onstage, and deliver.

6.6 Action items

For this chapter, I'd like you to focus on some of the professionalism rules I've covered in this chapter:

- Where would you paint your yellow line? What sorts of things would you expect to leave behind it? Make a list of those things. As you get started with the yellow line concept, having that list handy—on your phone, perhaps—lets you review it each day, making a conscious activity of crossing the line.
- Have you ever overcommitted and been unable to be your word about something? Try making a list of all the current commitments you have—to work and to your personal life—and evaluate your ability to meet each of them. Is now a time to renegotiate any that you might be at risk of missing? Can you use such a list to start getting a better idea of what you *can* do so that you can avoid overcommitting in the future?

I often keep a list of current work commitments in a note on my laptop. That way, when someone asks me to commit to something new, I can instantly pull up my list and ask, “Which of these can we drop?”

- Get yourself an antitrigger that forces you to slow down and pay attention to details. For me, it's a little fidget button I bought at a dollar store. It's a steel button that makes a soft click when I squeeze it. As I go through my day, I try to notice when I'm letting details slide by: the names of the stores I'm walking past, the traffic sign I passed, and so on. I click my button and deliberately slow down and look at those things as I do so. Over time, it's helped my brain stop filtering information. Now, when I need to concentrate on something, I can softly click the button, and it helps tell my brain to slow down and stop filtering. Details become clearer, and I'm often able to solve problems faster as a result.



Manage your time

Many technology professionals struggle with time management. Sometimes, our time is taken from us in the form of meetings and other obligations. But most of us can also do a better job of managing the time that is left to us, and learning to manage our time is a mission-critical skill for a successful career.

7.1 Discipline, procrastination, and laziness

I first want to briefly make some distinctions between *procrastination* and a behavior that it is often confused with: *laziness*. In my view, procrastination is the spiritual opposite of *discipline*. So let's start with my simple definition of discipline.

DISCIPLINE Remembering why you started doing something in the first place.

When I talk to people about why they delay doing some tasks but jump right into doing others, their explanation usually boils down to something like “I just didn’t feel like doing it right then.” And I understand this. I’m a writer, and sometimes I am very much not in the mood for writing. I’m rarely in the mood for reviewing edits (ask my editors). So it’s easy to put off doing these tasks—in other words, to *procrastinate*, to delay working on a project or finishing a task. Am I being lazy? Maybe.

But then I remember why I started writing the book in the first place. I had a goal, and I had outcomes I expected to achieve: fame, fortune, Pulitzer recognition, you know, the usual things <grin>. And I still want those things, so I need to get it done. **Perhaps this understanding of discipline suggests a good definition of laziness.**

LAZINESS Forgetting or not knowing why you started doing something in the first place.

I've definitely had projects that I put off and put off, and putting them off made me feel bad. Soon I felt bad about the whole project, so I put it off even further. Eventually, I sat down and realized I didn't have a good reason to be doing the project. I could not see how it fit into my life or career, and I could not identify what outcomes it was helping me achieve. In those cases, I dropped the project instead of putting it off further. If I couldn't explain to myself *why* I was doing it and what value it brought me, why bother stressing over it? I admitted that maybe it didn't need doing, cut my losses, and moved on.

You do need to be a little careful about jumping to conclusions in that regard, though. Don't write something off because you're not in the mood for it right now. Instead, sit yourself down in front of a mirror, look carefully into your eyes, and ask yourself, "Why did I start doing this in the first place? What did I hope to achieve by it?" Answer yourself and then ask whether your reasons are still valid.

Some projects may not have a reason to continue doing. I'll give you an example: podcasts. I've probably started three of them and abandoned them after 20 or so episodes. They're a lot of work. And I remember why I started doing them! In some cases, however, I decided I simply no longer wanted the outcome I'd originally planned. In other cases, I decided the podcast wasn't actually achieving the outcome I'd hoped for. So I dropped those projects.

But for those projects I do have a reason for doing, and when that reason is still valid, I have to manage my time so that I can get them done.

7.2 Time management

I wrote a bit about time management in chapter 6 because I've come to realize that **being an effective time manager is a key behavior of almost any effective professional**. In this section, I want to be a bit more prescriptive and give you some specific behaviors to watch out for and some specific actions to consider taking.

Time is obviously a finite resource, and being able to manage it effectively is an absolutely critical career skill. Let's start by looking into a few universal techniques that can help you become a better time manager in almost any professional situation.

7.2.1 Time inventory: The TimeFlip technique

Successfully managing any resource begins with understanding what you already have. In this case, I do not mean the amount of time you have but how you already spend your time. To keep track of how I spend my time, I use a TimeFlip device (<https://timeflip.io>), although you could easily do the same thing with a simple journal or a software app. I like having a physical device to use. A TimeFlip is a white plastic dodecagon—basically a large 12-sided die, like the one in figure 7.1.

You label each side of the TimeFlip with the various activities you normally perform during the day: coding, killing time on Twitter, going to meetings, getting

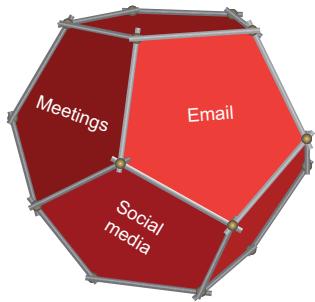


Figure 7.1 Think of a TimeFlip as looking something like this.

interrupted by someone tapping on your shoulder, whatever. A companion software app runs on your computer. Whenever you switch activities, you quickly flip the TimeFlip so that whatever activity you're now performing is at the top. You do this all day, every day. I did it for a couple of weeks before I looked at the results. I got something like figure 7.2.

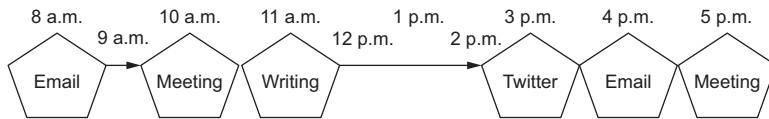


Figure 7.2 A TimeFlip baseline

What the TimeFlip did for me was help establish a baseline. The first two weeks I used it turned out to be reasonably productive in my estimation, and I felt that if every week could go about the same, I'd be satisfied with myself. So the TimeFlip showed me how much time I spent on social media during the day, how much time I spent in meetings, and how much time I spent writing PowerShell scripts and doing every other major activity that comprised my days.

Now remember: I'd already decided that in retrospect, those were good weeks. That meant I didn't need to beat myself up for using social media or going to the break room. I'd done all of those time-wasting things and still had what I felt was a good week. So those so-called time-wasting things were part of an ordinary, healthy work week for me.

Moving past my initial two weeks, I continued using the TimeFlip (and still do to this day). I measure each new week against my baseline. I'll notice in some weeks that I'm burning too much time on social media, which means I try to manage that better going forward by using time rationing (which I'll discuss in section 7.2.2). Some weeks have more meetings, and I can track that time and correlate it to any increases or decreases in my productivity. ("Yes, boss, I did write fewer scripts last week, but you had me spending 20% more of my time in meetings. Time is finite, and you get to decide where I spend it.")

Using the TimeFlip has been an incredibly useful exercise for me, and I update my baseline time expenditure two or three times a year (or when the nature of my work changes significantly, such as being assigned to a new project). Without it, I simply wouldn't know where all my time went. Now I do.

These days, I also tend to think of my days in ledger form. This helps me plan my day. (I use a Rocketbook planner, which uses plastic-coated pages and a special marker, so I can wipe off and reuse it each week.) A day's ledger might look like table 7.1.

Table 7.1 Day's ledger

Time block	Planned task	Actual task
8–9 a.m.	Email and Slack follow-ups	Email and Slack
9 a.m.–12 p.m.	Meetings and/or doc reviews	All meetings (doc reviews in the morning seem to never happen)
12–1 p.m.	Lunch	Lunch and postponed doc reviews
1–2 p.m.	Doc reviews, personnel stuff	All personnel stuff
2–2:30 p.m.	Social media	
2:30–4 p.m.	Meetings	
4–5 p.m.	Clear emails and Slack to-dos	

What's important is that I based this blocking of my time on what my TimeFlip told me I was already doing. I optimized a bit where I felt I could but didn't try to remove all of my wasted time, like the time I might spend on social media. My brain needs that time, so I'm planning for it. I'm also not trying to stick that wasted time at the end of the day; my TimeFlip indicates that I tend to start straying off topic in the afternoons, so I plan for that to continue happening. This ledger shows me when I've got time for meetings, so I try as much as possible to fit my meetings into those blocks. This system doesn't always work perfectly, but it helps me go into the week with a plan versus reacting to whatever comes up.

Note that my ledger has room for me to note what happened! This lets me change and adjust as I go, and at the end of the week, it lets me see how successful I was at staying on plan. Knowing what happened helps me make next week's plan more accurate.

7.2.2 **Time rationing: The Pomodoro technique**

Francesco Cirillo invented this technique (<http://mng.bz/O1wP>), which I use to plan and ration my time. His idea was to block your out into defined chunks, assign work to those chunks, and work on each chunk only for its predetermined amount of time. To keep himself on track, he used a mechanical kitchen timer that happened to be shaped like a tomato—hence the name of the technique, as *pomodoro* is Italian for *tomato*!

Francesco worked in 25-minute chunks of time (which he called *pomodoros*), separated by 5-minute breaks. The idea behind this approach is twofold: you can get a lot

of work done in short but focused intervals, and frequent short breaks keep you from mental and physical fatigue. I'll adjust my chunk size depending on the nature of my work at the time. I've used 25-minute pomodoros, 55-minute ones, and other intervals that better match the nature of what I'm up to.

It works like this: you assign a task to a chunk of time and then set a timer. You work on that task until time runs out. Then you take a break (I time my breaks too) and move on to the next task. I used my TimeFlip data to help me estimate how long a given task is going to take me, and I let my work calendar fill in a lot of the spots. I might have an 8-hour day, **broken into 16 25-minute pomodoros, that looks like this:**

- 1 Email/Slack catch-up
- 2 Wander around the office and do personal catch-up time
- 3 Scheduled team meeting
- 4 Scheduled team meeting (continued)
- 5 Review unit test logs from last night
- 6 Plan time chunks for tomorrow, based on email, team meeting, and unit test failures
- 7 Lunch
- 8 Lunch (continued)
- 9 Performance review with boss
- 10 Coding time
- 11 Coding time (continued)
- 12 Coding time (continued)
- 13 Coding time (continued)
- 14 Cross-team standup meeting
- 15 Check in, deploy, and run unit tests
- 16 Email/Slack catch-up (4:30 to 4:55)

Not every single one of my days is planned to that level of precision, but many days are. Some days, for example, I plan to work on a single project, so breaking it down into chunks doesn't gain me anything. But when I know I plan to shift between many things, planning helps me stay on track.

By starting with communications catch-up, I can quickly see whether I need the day to be a multitasking day or whether I'll have time to get some single-task work done. I've got plenty of downtime for my brain (80 minutes total, if you're counting), and those breaks make it easier for me to switch contexts into the next task (while I grab a Coke).

Not every day goes perfectly, of course. But when things do go off course, I can see the consequences and plan for them. ("Hey, boss, I know the meeting ran long, but that means I'm either going to have to cut coding time back by half an hour, or skip the standup this afternoon. What works best for you?")

I've worked with colleagues who've taken this one clever step further. They work in open-plan offices, where distractions are a constant fact of life, so they bought large-digit

digital countdown timers. They mounted those timers so that they were clearly visible to anyone walking by, along with a sign that said “Minutes until you can bug me.” Their office neighbors quickly accepted the convention. Knowing that you could tap someone on the shoulder in less than 25 minutes made it easier to wait and in fact encouraged more people to send an email, which was dealt with during its designated time.

7.2.3 **Time catalog: Knowing your capabilities**

My extensive use of my TimeFlip and a kitchen timer eventually helped me realize a lot about myself. I know I can productively code for about 90 minutes before I need to switch tasks (the reason I’m not a professional software developer anymore). I can write for three or four hours solid. I do better with 25-minute meetings than 50-minute ones. It takes me about 20 minutes to clear my overnight emails and about 25 minutes to handle the urgent Slack messages of the day. I know I need about 75 minutes of “no-thinking” time during the day, which I’m fine consuming in small chunks along with a bigger break while I eat lunch.

These facts help create my time modules. I have a catalog of them now, representing a list of common things I do and how long I need to do them. I use that catalog to inform how I chunk out my work days. I know there’s no point in scheduling a 25-minute writing block; I need three or four hours to be productive. I know I don’t need to schedule an hour for emails every day; half an hour is fine. I know that if I’ve scheduled three chunks of time to code and something interrupts, I might as well forget about any coding that day because I won’t be able to stay in the groove long enough.

Knowing these time-based facts about myself has informed my time management efforts and enabled me to manage my time rather than let it get away from me.

7.3 **Multitasking**

Do you multitask well? For the sake of argument, say—out loud—“Yeah, I multitask pretty well.”

No, you don’t. There is no such thing as multitasking. This word comes to us, in modern times, from computers. Computers do not multitask either. Not really. For a given microprocessor (we’re talking about a core here, which represents your brain in this analogy), you can only do one thing at a time. True multitasking can occur in a multiprocessor computer, but that’s (in this analogy) like having multiple brains. You don’t. You’re a single-brain machine; therefore, you can only do one thing at a time.

Computers provide the appearance of multitasking by rapidly switching between tasks. This process is called task switching. The theory is that a computer spends most of its time waiting on humans, so in the idle times, it goes off and does other things. Computers are great at task switching because they have perfect memories; they never lose track of where they were and what they were doing.

Humans do not have perfect memories. We lose track of where we were and what we were doing, so we are also imperfect task switchers. What we can do, however, is

learn to recognize idle time and switch tasks then. This is what most people are not good at, because they like to think that multitasking is a real thing. Recognizing that you've entered an idle state ("I'm waiting on Jason to get back to me on that email") lets you switch tasks to something else you can work on. If you have exceptionally good short-term memory and deep attention to detail, you can hop back to the original task when it's no longer idle, creating the appearance of multitasking.

I can task-switch like a machine when I need to. I can track multiple conversations, to-do lists, contexts, and more in my head. I can be extremely productive—in a certain kind of way. By this, I mean that if my tasks of the day are all short, sprint-type units of work with a lot of idle time (like collecting information from colleagues and making decisions on that information), I can get a lot done. But that isn't the only kind of work I do. When I sit down to write, I can't task-switch, because writing doesn't offer idle time. There are no natural stopping points, and anything other than the writing serves as a distraction for me. **When I'm writing, a distraction forces me to switch from a task that needs my attention, resulting in less productivity. So when I'm not in task-switching mode, I turn off the distractions: close the door, turn off pop-up notifications, even cut off my Wi-Fi if I can. Only a high-level, urgent interruption, such as a phone call, can force me off task when I'm in single-task mode.**

The problem is that most people don't deliberately manage the mode they're in. **They don't put themselves in single-task mode**, and they don't recognize that multitasking mode—really, task-switching mode—is a distinct and different way of operating. They don't schedule themselves to work in those various modes; perhaps they multitask in the mornings and single-task in the afternoons. But if you can learn to recognize these distinct work modes, you'll be infinitely more productive, more satisfied, and more sane.

Here's how to avoid being tripped up by multitasking:

- Recognize when it's time to put yourself in single-task mode, such as when you're coding, writing something, or working on something else that needs your full attention for a long period. Do what you can to reduce distractions that will take you off task. If you work in an office, try to come up with an agreed-upon signal, such as "**If I'm wearing headphones, please leave me alone.**" Shut down your email app, Slack, and other electronic distractions.
- When it's time to be in multitasking mode, don't dive into work that will require your attention for more than a few minutes. Instead, take care of numerous tasks that you can get done quickly. **Take notes, if needed, to help you remember where you were in each of the tasks you're juggling, and keep a list of any action items that you're responsible for handling.**
- Don't mix modes. If you get into single-task mode, but the realities of work demand your attention, put your single-task work away and **bow to the necessity** of working in multitasking mode for the time. Come back to your single-task work when you can safely block off sufficient time.

7.4 Action items

For this chapter, I'd like you to look at what's on your plate and start to apply some time-management exercises:

- Start by inventorying all the projects you have in your life, which can include work projects as well as a to-do list of home projects. Do you know *why* you're doing each one? Can you clearly state the outcomes you expect from each and explain the value you believe each one will bring? Consider eliminating any for which you can't.
- **Create a time catalog for yourself.** Begin by tracking how you currently spend your time, with no judgment. If you spend eight hours a week watching cat videos on YouTube, fine; document it. Do this exercise for a week, and try to identify days that you consider normal or acceptable for yourself. This will help you understand where and how you normally spend your time, and help you identify time that you can put to other uses.
- Try a time management technique like Pomodoro, and see whether it feels effective for you. It may take a few weeks of effort for things to become routine, so don't worry if you're not following the technique strictly at first.

Handle remote work

As we've all learned, the chances of anyone in tech eventually having to work remotely even part-time are nonzero. **Remote working can be challenging**, but it's definitely a skill you can build. And because there's always the possibility of it being something you need for your career, why not plan for it and start building the skill now?

8.1 The challenges of being remote

If you're not used to working remotely and haven't developed a system for doing so effectively, know that remote working—whether full-time or occasionally—can definitely be hard on your work and on your career. Frankly, it can also be emotionally draining.

It becomes easy, for example, to lose sight of the yellow line that delineates your personal self from your professional self (see chapter 6). When you're working from home, it's a lot harder to create a firm line between work life and home life, which sometimes makes it difficult to put on your office face and show up with all the professionalism you know you should. It can also be easy to work too much. I've worked from home for years, and I sometimes run into folks who think it sounds great until I rephrase it as "I sleep at the office." I've learned to maintain that work-life balance by using the time management techniques I discussed in chapter 7.

Additionally, remote working can feel isolating, potentially making you unhappy if you normally thrive on daily office interactions. There are no break-room conversations, no random hallway encounters with fellow employees, and no opportunities to run across people outside your team. If the majority of your colleagues are in the office, and you're remote, it becomes easy to be left out and to feel left out even when you aren't. When you're not visibly present at the office every day, **you may**

feel that you're being passed over for promotions, choice projects, and other internal opportunities.

Even participating in video calls as a remote worker can be draining. Video calls don't convey all the subtle human body language that we've evolved and grown up to expect, making the people we're talking to seem flatter and communication more difficult. Your brain winds up exerting additional energy to stay engaged in conversations and to perform all the subconscious signal reading that we normally do effortlessly in person.¹

Remote work can also be tough on your family. Young children don't always understand that even though Mom or Dad is home, they're not necessarily accessible. The confusion can be frustrating for the kids as well as the parents.

In short, remote working isn't like working in an office at all. If you're accustomed to working in an office or if you're generally new to the workplace, remote working can be far more difficult. I consider myself an introvert, yet working entirely from home is more challenging most of the time than working in an office. So what can you do to make remote working part of your success?

8.2 **Creating a space**

The first step is creating a space that's used for work and nothing but work. Ideally, this space is a dedicated room in your home with a door you can close. That's not possible for everyone, so at the very least you need a dedicated and separated space of some kind. I have a friend who now works from home and doesn't have a spare room to use as an office. Instead, he set up a desk in a specific area of the house and put bright yellow carpet tape down around it. That's his office. He never sits inside the square unless he's working, and the rest of the family has learned to respect the invisible wall that the tape represents.

You cannot be successful over the long term treating your couch and coffee table as your office. You need a space that gives you many of the things that an office environment provides: a place to sit where you can hold an appropriate posture, and room for your work equipment and for whatever accessories you need: notepads, stress balls, or whatever else gets you through the day at work. Even working at the dining table will become stressful over the long term. Either you've permanently taken a space away from your family and made it into your office, or you're left constantly clearing out of your office every afternoon and moving back in every morning.

A SPACE IS ESSENTIAL If you don't have the ability to create a sufficient, dedicated work space for yourself, you run a real risk of not being successful as a remote worker. Take that into account as you consider both housing options and job opportunities.

¹ In "Zoom Gloom" (<http://mng.bz/YAZ7>), Julia Sklar explains how video calls affect us biologically and psychologically.

Additionally, if you work from home, you by definition sleep at the office, and that's no fun. Make sure you're getting *out* of your work space every day and maintaining a healthy separation between work and life. This is a big reason why having a dedicated, set-aside work space is so important. But it's also important that you get out of the house for a walk or run, to walk the dog, or to meet with friends. I have a friend whose main hobby used to be playing games on his Xbox; when he was forced to begin working at home, it resulted in too much time in the same place. He dropped the Xbox games for a long time and took up herb gardening in his backyard. It gave him the change of scenery he needed and a place he could walk away to when he needed a break from work.

CONSIDER OFFICE STRESS Work can be stressful; we all know that. Most of us rely on being able to leave and go home, where we try to create a more comforting, less-stressful environment for ourselves. Imagine what it's like to have all that stress occurring at home, with nowhere to get away to! That's one more reason why a dedicated work space can be so critical and why it's so important to be able to get *out* on a regular basis.

Although everyone needs a dedicated work space, we don't all need the same kind of space. I need a full-on office with a door that I can close. (I'm an obnoxiously loud typist, and my family needs the door more than I do.) One friend of mine gets by on a little desk that folds out from a wall-mounted cabinet. Yet another has a corner in her basement where she feels happy and productive. Still another pays—out of his own pocket—to rent a small enclosed space at a co-working facility near his home. There are plenty of options to consider.

Everyone is different when it comes to creating an effective workspace, and that can include your electronic workspace. Maybe your employer sends you a laptop to use while working at home, but you find it more convenient to also have that mail on your personal phone. That's fine, but be aware of how easy it becomes to break the work-life balance and wind up being always on for work. There's nothing inherently wrong about being always on for work, provided that you've made a conscious decision to be that way, but don't let it happen by accident.

8.3

Creating a space when you have no space

One serious challenge of working remotely is not being able to create a dedicated space for your work. Commandeering the dining room table, working on the sofa, or perching on your back patio can present a lot of potential problems:

- Working in those shared spaces can create difficulties for your family (something I'll try to address in section 8.4).
- Not having a dedicated space makes it harder to step away from work and go back to your personal life. The two can become inextricably linked, which means you feel like you're always working and never have any time away from the office. That's going to put a lot of stress on your mental health.

Although I haven't met all the people in the world, I have never met anyone who was able to successfully work remotely, for a long, sustained period who didn't find some way of creating a dedicated space for themselves. I know people who tried, and they became stressed, unhappy, and upset with their jobs—something that I certainly hope never happens to you. With that in mind, here are some suggestions for more creative ways to make space for work:

- **Consider renting space.** Even if you don't work in a rented space every day, spending a good chunk of your workday *elsewhere* can be helpful. Co-working spaces often rent single, dedicated offices by the day, week, or month, along with access to open-plan workspaces where you can have your own desk. Some areas of your town may offer affordable office-space leases as well. Where I live, the downtown area has several older homes that have been converted to office spaces, with what used to be a bedroom renting for pretty low monthly rates.
- **Look into innovative furniture.** I have a friend who lives in a one-bedroom apartment. She didn't want to turn her dining room table into her desk because there was no way to get away when the work day was done; her desk was right there! But she realized that she didn't spend much time in her bedroom during the day, so she invested in a kind of Murphy bed. When she wakes up, her bed folds up into a wall-mounted cabinet, and a desk swings down from the underside of the bed. It's got plenty of space for her laptop, and she taped a plastic folio to the underside of the bed as well, which gives her a spot to stash her notebook. When the work day is over, her laptop goes in its bag, the bed folds down, and she heads into the rest of the apartment to enjoy dinner, watch TV, and relax.

Another friend converted a wooden shed to an office. He installed a small air conditioner, cut a window in one wall, and reorganized the things that were being stored in the shed. (He admits to having had a sizable yard sale for all the stuff they decided they didn't need as much as they'd once thought.) It's given him a private place to get away to and a way to walk away at the end of the day.

The most creative approach I saw was taken by a friend whose spouse also worked from home. The spouse had always worked from home, so they had a small bedroom that served as a home office. They tried sharing that space, but it didn't work out for either of them. So my friend bought some ceiling tracks—the kind you see in hospitals to separate a room into halves. She hung some attractive curtains from the ceiling and pulled them around part of the living room during the day. She got a coffee table that could extend upward to desk height, and that became her desk. After work, the curtain tucked out of the way next to the sofa, the laptop went into the coffee table, and the desk lowered back to coffee-table height.

The idea is to be creative. Look for spaces in your home that may be underused during the work day, and look for furniture and other solutions to help turn them

into an office. Don't forget about your yard, if you have one: accessory structures can have a small space heater or air conditioner added to make them more livable. Even an enclosed pop-up tent (like the kind you often see at farmer's markets) can offer a way to separate your work and personal lives a bit, at least now and then.

If you think remote work might be more permanent for you, it might be time to start thinking about changing residences. A bigger apartment, or a home with a dedicated den or office space, might be a good long-term bet, provided that it fits your finances.

8.4 **Working with family**

Perhaps the toughest part of working from home is sharing the home with people who don't fully understand what you're doing there all day. My own mom, when she would visit us, had this problem: even though we were lucky enough to have an in-law suite with its own bedroom and bathroom for her, she saw nothing wrong with popping into the spare bedroom I used as a home office. After all, I was "playing on my computer" like I did when I was a kid. What harm would an interruption cause?

The trick is to set expectations for others and lower expectations for yourself. If your job is one that routinely expects you to be in back-to-back meetings, that probably isn't going to work. You're going to need to schedule more down time so that your kids can see you.

One colleague has a giant whiteboard on the door of the room she works in, with a battery-powered clock hanging right above it. Her meetings are clearly mapped out each day, with a chunk of time between meetings. "My kids can wait 30 or 45 minutes," she told me, "provided they know they'll get 15 or 20 minutes of my attention afterwards. Honestly, they're usually bored of me after 15 minutes, so I can take another meeting until they think up something else they just have to share with me." She put me onto an online app called Clockwise, which integrates with Google Calendar and grabs unscheduled time on your calendar as focus time, blocking that time out and letting you tend to yourself—or your family—throughout the day.

After you set those expectations, you have to honor them. If a meeting runs long, excuse yourself, and explain that you have another meeting. And you do: it's with your family. Communicate your days off, and schedule those too: make time for chores, but also make time for the family to be a family.

You and the other adults in your life may need to have a frank conversation about changed expectations and agreements, if you find yourself working from home. Adults at least, can see logic and reason where younger kids sometimes can't. Talk about the realities of work and about what still needs to happen at home, and create written schedules and agreements about the new way things will need to work. I have one friend who found herself working from home, and her partner kept wanting to vacuum when she was on meetings. "And I get it," she told me. "That's when he's used to cleaning. We had to shift that to the evenings and weekends, which took some adjusting, but he moved dusting to the afternoons, and we

agreed to buy a second vacuum so I could help get that done faster in the evenings. And honestly, it's half an hour of not having to think about anything, so it's a bit of an unwind after work."

Be creative, and think critically about the problem. Don't expect adult behavior from your kids, who are used to having their parents pay full attention to them whenever they're home. Don't expect a spouse who's normally been at home when you were at work to immediately rearrange their world to accommodate yours. Sit down, talk through the situation, and come up with a plan.

8.5 **Adopting a routine**

Most people who work in an office have a regular routine: they commute to the office, perhaps listening to the radio or a podcast along the way. When they get to the office, they might grab a coffee from the break room and sit down at their desk to work on the day's mail. They do their job, taking meetings as scheduled, and then head home. Perhaps they have a habit of stopping to pick up groceries or a meal on the way home. When they get home, they might watch the evening news or a TV show, and spend some time with the family.

What the routine is doesn't matter; what matters is having a routine. That's especially true for a remote worker, because it helps enhance the mental separation between work life and home life.

A colleague of mine has established what I think might be a model routine for remote working—not the contents of her routine necessarily, but the structure she's put into it. She wakes up, has a cup of coffee, and gets the kids off to school. Then she takes a jog around her neighborhood. This jog is her "commute." She listens to her podcasts each morning and starts to transition from home to work. She gets home, showers, and dresses in work clothes. Then she sits in her office space and works. She schedules her lunches, blocking them off on her calendar. Her employer embraces remote work and has a tablet set up in each break room that's dialed into a perpetual video call. When she's ready for a break, she uses her phone to dial into that call and heads to her kitchen for a snack. That way, she's able to chat with anyone who might be in the break room, as the people in the office do. When the work day is over, she takes a quick walk around the neighborhood. It's her commute home, and she says it's when she unwinds and relaxes, listening to music she enjoys. The kids usually get home from school before she's done with work, but they know that Mom isn't to be bothered until after her walk.

One thing I admire about her—and something I emulate—is how rigorous she is about scheduling her day. Everything goes on her calendar, and she lives by it. Her day doesn't have the spontaneity of a normal office environment, where random conversations are more likely to happen, but she's also convinced many of her colleagues to strictly schedule their day. When they all get to work for the day—some in the office, some remotely—one of their first tasks is to look for blank spots in their calendar. They'll fill those blank spots with 10- or 15-minute meetings with other colleagues who

also have a blank spot then. Essentially, they're duplicating the random encounters of the office more deliberately.

Whatever you put in your routine, the point is to have one and to stick with it. It'll help keep you sane and help you show up as a professional every day.

8.6 **Explicitly defining a culture**

While it's up to the entire company to create and promote a culture that embraces remote work, every remote and in-office worker can do a lot to help drive that culture. Meetings in particular can be challenging for remote workers, especially when they're in a minority. It's important that you help create a culture that embraces remote work and affords appropriate courtesies to both in-office and remote workers:

- Video calls often have a delay of a second or two. Everyone in the organization should understand that, and if they ask something like "Does anyone have anything to add?", get used to looking around the room for a few seconds. That gives remote people a chance to respond.
- A noisy conference room is the worst experience for someone who's dialed in. Side conversations, rustling candy wrappers, and tapping pens can make the conversation incomprehensible. Meeting facilitators need to recognize that and remind everyone who's in office to mute themselves, as the remote attendees are expected to mute themselves to reduce background noise.
- Some companies I've worked with use the talking-stick mechanism to help remind people to be quiet. They'll use some physical token—maybe a stress ball or some other object—to indicate who may speak. For the in-office people, it's a visible reminder to be on mute when you don't have the object. When a remote person is speaking, the object is placed in a small container, like a cup, to indicate who is holding it at the moment.
- Remote workers are asked to behave as though they're in the office. That means dressing according to office customs or rules, showing up promptly to meetings, and maintaining a professional work environment (even though it's in their home).
- It's best if meeting facilitators show up a few minutes early so they can make sure that the conferencing equipment is set up and ready to go when the meeting is scheduled to begin. Meetings should always begin with short introductions or at least a greeting to and from everyone, as a way of making sure everyone can hear and can be heard without having to go through the painful "Can everyone hear me?" ritual every time.

Another area to consider is the inequity between in-office workers and remote workers when it comes to casual conversations. Everyone can, and should, be on guard for casual office discussion and decision-making that excludes remote workers. I once contracted with a company that had a unique take on the problem: rather than grouping teams in their office, they deliberately spread out. A programmer might sit next to

a clerk from finance, and a network engineer might sit next to someone from human resources. The idea was to require a more deliberate effort to walk over and talk to a colleague, reminding you that you might be excluding a remote colleague. Teams could meet in conference rooms, of course, but doing so is an obvious reminder to schedule something and include your remote colleagues. The solution wasn't perfect, but it went a long way toward helping in-office workers develop some empathy for their remote colleagues.

8.7 Networking like you're in the office

One of the most difficult things newly remote workers deal with is missing out on all that office chatter. Especially in companies whose leaders have not created an explicit remote culture, it's incredibly easy for remote workers to feel left out. Here are some suggestions you can consider to help feel more connected and more like part of the gang, even if the entire gang is remote:

- *Hold regular happy hours over Zoom, Teams, or whatever communications platform your organization uses.* Hold these events after work hours and for an hour or two. Focus on small groups: your immediate team, your direct peers, or similar groupings. Forbid work conversations. Focus on what everyone's doing this coming weekend, what they did for their birthday, and other personal conversation.
- *Keep your work calendar rigorously updated.* Mark everything, even personal appointments like lunch. Then tell your co-workers, "Hey, if you have a quick question, glance at the free/busy time on my calendar. If I'm free right then, send me a Zoom link (or whatever). If I don't respond within a minute, text me your question, but that way, we can connect in person more often, even if it's just for a minute."
- *Look for tools that help co-workers connect in person at random.* Slack has a free plugin called Donut (<https://slack.com/apps/A11MJ51SR-donut>) that's designed to randomly connect co-workers for a quick video chat or in-person meeting. It's a good way to have a human connection outside scheduled meetings. You can talk about stuff you're working on and get a better feel for what the rest of the company is doing.

Networking with non-co-workers is just as important. Look for open Slack groups within your profession, and attend lots of virtual events. It's not the same as coffee or drinks with colleagues, but it still helps keep you in the mix. A little search engine research for terms like "remote angular slack group" can turn up several options for folks who work with the Angular framework, for example, and you can tweak that search for your own interests.

8.8 **Remote work: Permanent or temporary?**

After the COVID-19 pandemic of 2020, a lot of us found ourselves working remotely. What a lot of us didn't know—and in many cases still don't—was whether remote work would last forever. Although some companies are now reopening offices in parts of the world, and others have announced plans to do so eventually, some of us still don't know. Some of us have taken advantage of the new remote-friendliness of the world to move to different cities, to save money, be someplace we love, be closer to family, or for some other personal reason. For many of us, then, remote work is definitely not temporary.

Don't let your temporary remote accommodations accidentally become permanent. Something that was bearable for a few months or a year may be wearing on you in ways you don't sense until it all finally becomes too much. If something is going to go on forever—or even for the foreseeable future—sit down and think about it. Examine all of your assumptions about life and work: how much space you need, where you work, what kind of work you do, how your family interacts with you, everything. Put it all on the table, discuss it with the people who are important to you, and thoughtfully make a plan. You need to be willing to set aside all your assumptions so that you can view your situation with a fresh eye, consider your own unique needs and boundaries, and come up with a best-fit solution.

8.9 **Action items**

For this chapter, whether you're a remote worker or an in-office worker, I want you to think about remote working life. Create a sort of remote work preparedness plan, or if you're already remote, evaluate your current working conditions:

- What kind of space would you create to work in? Would you be able to meet some of the criteria I outlined in this chapter?
- What kind of work-from-home routine would you create for yourself? What would your morning commute look like, for example?
- If you were to draft a list of office etiquette rules to make remote workers feel more included, what would that list look like?
- If you've never worked remotely (or have done so only infrequently), what sorts of things would you miss about being in the office? How could you mitigate the loss of those things by using tools, processes, or rules of etiquette?

Be a team player

“Can’t you just let me write code and leave me alone?” It’s a question I’ve asked more than once (along with “let me run the servers,” “let me build the DevOps pipeline,” and related variants), and it’s a question whose answer is invariably no. Teamwork is an integral part of getting things done in tech, and your ability to be an effective, positive teammate is probably one of the top skills any employer looks for.

9.1 *The ups and downs of teams*

I will freely admit, now that I’m a bit older and it’s in my past, that I haven’t always been a great team player. If you’ve seen *The Big Bang Theory* on TV, I’m Sheldon Cooper in far too many ways. You might be a more natural team player, and some of the advice in this chapter may seem somewhat obvious to you. But if you’re like me, the tips I’ll present are things that you have to actively focus on every day and put real effort and energy into.

One of my weaknesses stems from a strength: I’m a “Get things done” worker. Point me at a problem, get out of my way, and be prepared to bury the bodies I leave in my wake. In meetings, I’ll state my views clearly, and I’m often the first one to speak up. That’s the strength. But then I’ll then defend those views to the death, convinced that my way is the best way forward. I want everyone on my team to be on my side and to rally for my idea.

I’m really not like that anymore, but when I was younger, I was probably a bit of a jerk. It took me several years of working as an independent contractor—I literally was the team—to realize how much more a functional, effective team can accomplish than any individual can do on their own. I’ve always been a little sensitive, and

it's pretty easy to hurt my feelings, but it took me a long time to realize that other people can be just as sensitive and have their feelings hurt just as easily. That realization was the beginning of my being a better teammate, and I'd like to think that at this point in my life, **I'm a lot better at it.**

Great teams can accomplish amazing things in technology. I've worked with teams that created world-class certification exams, designed and delivered amazing software products, and produced groundbreaking community events and charitable outcomes. But being on a team means you sometimes have to moderate parts of yourself. You **can't be afraid to speak up when you have something valuable to offer, but you can't do so in a way that suppresses other people's opportunity to contribute.** You have to bring your ideas and experience, but you have to **offer them, not impose them. You have to acknowledge that even if you are absolutely sure your way is best, it might not be the best way for the team to proceed.**

Teams are made up of people, of course, and people can be both a **joy and a trial.** **We all bring baggage**, both good and bad, to our teams: our biases, our experiences, our failures, our preferences, our successes, our prejudices, and more. What we bring often conflicts with what our colleagues bring, and conflict is where teamwork can become really challenging. But properly understood and managed, conflict can also be where inspiration and innovation come from. The best teams don't try to suppress or avoid conflict; **they embrace it** and use it in healthy ways.

So this chapter is all about being a better team player. As I've said, this activity doesn't always come naturally to me, although I've definitely grown and gotten better over time. I still have to work on teamwork, putting energy into it every day, and the best way for me to do that is to have a sort of behavioral checklist. It's something I can review every day (I do, to help keep myself in the best groove possible for the teams I work on).

9.2

A checklist for being a better team player

Here's my daily checklist for being a better teammate:

- **Know why I'm here.** I make sure that every day, I am reminding myself what the team's mission is and what my role is on the team. Role is hugely important: I need to respect the fact that my role owns certain tasks and responsibilities, just as other teammates' roles own specific responsibilities. I can make suggestions to team members in other roles and accept suggestions for my role, but ultimately, we all need to respect what we each were hired to do. In my role as a DevOps engineer, for example, I often have some firm opinions about programming languages. Some languages are easier to deploy through a DevOps pipeline—they have better unit-testing frameworks, perhaps, or they create more self-contained packages—and I tend to like those. But although my role can offer those opinions and considerations, it's usually the software developers who get to choose their language. **They need to consider the suitability of the language to the task at hand, their own ability to code and maintain, and other**

concerns. So long as we're all keeping the team's overall mission firmly in sight, **we each need to respect the boundaries of our roles.**

- **Support failure.** Humans learn best by trying things, and we often fail a lot before we figure out the best way to do something. On a team, part of my personal mission is to **make sure my teammates feel safe in those failures**. Rather than assign blame, I want to help my team move right into the postmortem, where we can discuss what we did wrong and what we can learn from that.
- **Communicate respectfully.** **Express, don't suppress.** It's important that I bring my experience and perspectives to the team; that's part of what I'm paid to do. But it's also important for me to make room for others on the team to do the same. **I need to nurture a safe environment.** If someone on the team isn't communicating well—phrases like "That's a stupid idea" are a pretty clear indication that the environment isn't safe anymore—I need to step up and bring the conversation back to a healthier pattern.
- **Be committed as a group.** Many of the teams I've been on have had weekly or biweekly meetings, where we can update each other on our status and review our overall progress. I try to start those meetings by asking if we can quickly review our current goals and mission. It's a way to get everyone on the same page and make sure that all **subsequent conversation** is focused on achieving that shared mission.
- **Resolve conflicts within 48 hours.** I'm a big fan of taking some cool-off time if I'm angry about something one of my teammates did or said, but I won't give myself more than a couple of days before I address the conflict. Chapter 13 shares the context-seeking methodology **I use to try to defuse situations, understand my teammates better, and get conflict out in the open and resolved.**
- **Ask for help.** Showing vulnerability is important because it communicates to the rest of the team that I don't expect them to know All the Things, All the Time. I want my teammates to feel they can ask for my help whenever they need it, so I make sure I'm asking for their help as well. Even if I'm asking someone to double-check some code or make sure I'm thinking of everything during a server migration, **I want them to know that I need them.**
- **Get a little personal.** This isn't something that comes easily to me, because I'm a private individual. But I try hard to share a little bit about my personal life—who my family is, what we enjoy doing in our free time, and what we might be struggling with. I want to know the same about my teammates. All that sharing humanizes us. The less anonymous we are, the more easily we can accept our faults and personality blemishes, and the more effectively we can work together.
- **Help my teammates succeed.** **I spent a lot of years focused on my own success**, but what I've realized is that I'd rather be measured by the people I've helped succeed. I ask my teammates to share their definition of success and ask how I can help them realize that success. They almost always reciprocate, meaning that we're all helping each other succeed. In addition to accomplishing the missions

of our organization, we're able to help one another accomplish our own personal missions.

- **Stop and listen.** I was raised with an engineering mindset: when I see a problem, I leap to solve it. What I've learned is that my team and I can usually create better solutions together when I stop and listen **before leaping**. I'll ask my team to help me confirm my understanding of the problem, and we'll brainstorm solutions together. Everyone on a team occasionally needs a win—a time when they came up with the idea that solved everything. If I want people to give me those wins now and again, I need to work really hard to give them to other people as well.
- **Follow.** I'm a bit of an alpha personality, which you've probably figured out by reading this far. But on a team, I sometimes need to actively be a follower. Even on teams where I'm the nominal leader, I sometimes need to step aside and let others take the lead. My followership skills aren't always strong, and this is an area I constantly work on, but it lets other members of my team have a chance to shine, and I find that they never disappoint.
- **Understand your tools.** I've found a lot of value in sitting down with my team once a year or whenever our team's membership changes and discussing what each of us does well, what each of us doesn't do so well, and what each of us hopes to get from working on a team. After these discussions, my teams have sometimes approached our leaders and asked for our roles to be shifted slightly, because we've realized we weren't taking best advantage of our individual strengths. Understanding my team—what each person loves about their job, where they'd like to grow, how they prefer to communicate, and other individual eccentricities—has almost always made us more effective and happier.
- **Be positive.** Acknowledge people and their accomplishments rather than gossiping. Smile. Randomly tell people that you appreciate what they're doing and you hope they have a great day. See problems as opportunities, and express excitement about engaging those opportunities. Embrace differences rather than seeing them as "right" or "wrong." Show up to work every day, and be the kind of person who lights up your team rather than someone who turns them off.
- **Do the dirty work.** I've always been clear with my teams that I'm not too good to do anything. I know we sometimes worry that if we take on all the dirty jobs, everyone else will let us, and we'll be stuck with it. If your team is doing that, it's not a healthy team. Work to fix it or move, because on healthy teams, when I volunteer to do the jobs nobody likes, it often gets everyone in on the game, and we can share the workload.
- **Ask how I can help.** I try to ask this of everyone on my team at least once a day. Whatever they're working on, I'd like to know if I can make it easier, faster, or whatever. We have a shared mission, and I want to pull as much of that weight as I can with them.

- **Follow the Platinum Rule.** The Golden Rule, of course, is “Treat others as you’d like them to treat you.” But I’ve discovered that I don’t love that rule. I don’t really know what to do when other people tell me, “Hey, I really appreciate you, and I hope you have a great day.” But I’ve come to realize that they’re following the Golden Rule and treating me like they want to be treated. So for me, the Platinum Rule is “Treat others as they’d like to be treated.” Following that rule requires some careful observation and often requires me to flat-out ask people how they’d like to be treated. But it’s worth the effort, because in the end, we learn a lot more about how to be effective teammates.
- **Reflect.** Finally, at the end of each day, I spend some time reflecting on how I worked with my team that day. Did I do or say anything that was less than positive or that didn’t help move the team toward its goals? In other words, did I show up in the way that I wanted to?

9.3 Dealing with less-effective teams and teammates

I have a kind of escalation procedure when I find myself dealing with a team or a teammate that isn’t working for me:

- 1 I try to get really introspective and decide how much of the problem is on my end and how much is on theirs. It’s never 100% them; there’s always at least a little bit of a difficult situation that comes from me. If someone is using language that upsets me, I might ask myself, “Am I being oversensitive?” To check, I can look at how other teammates are reacting; if they are reacting negatively, this suggests that the problem isn’t entirely my own sensitivity. Finally, I try to understand the intent. Is the teammate’s language malicious or simply creating unintended consequences?
- 2 I try to talk through the problem by asking for context and by sharing my own context (a process covered more fully in chapter 13). “Hey, I was just wondering what’s going on with you today. Do you feel in a good mood? Anything bothering you? I wanted to share a little bit about the language you were using; I’m just wondering what that language means for you, and how you intend for other people to take it.” Creating a shared context is often the fastest path to a resolution or at least to removing misunderstanding.
- 3 If my context-sharing efforts don’t create the outcome I’d like, I’ll ask for help. I’ll start with other teammates. “So-and-so is using this language that really puts me on my guard. Is this just me, or is that something you’re feeling as well? Do you think the two of us could discuss it with them?”
- 4 I escalate to my team leader, presuming that my efforts up to this point weren’t successful. I’ll rarely ask them to solve the problem; instead, I ask for help and advice on solving the problem myself. After following that advice, I return to my leader and share the results. If I’ve been unsuccessful on my own, I might ask them to step in and help facilitate a conversation, with an eye toward creating a positive outcome.

- 5 I've come to this point only a couple of times in my career (thank goodness): I may need to consider leaving the team. That might involve an internal transfer, or it might involve a new job. If I've truly determined that a team or teammate problem isn't entirely in my own head, if I've been unsuccessful in solving it myself, and if my leader hasn't been successful or willing, it may be time to go. Life's too short to spend a third of it in a workplace where you can't be who you want to be and achieve what you want to achieve.

I do try to be as nonconfrontational as possible when seeking to resolve problems. I'll try to be verbally clear that the problem might simply be a misunderstanding on my part and that I'd like to resolve it. I may or may not believe that, but it lets me start the conversation without a direct attack ("You said something I don't like"). Repeating my understanding of the problem and offering the opportunity for someone to correct me is often the easiest way to move a conversation to a healthy place right out of the gate.

I'll also consider the health of my team. Do we have a clear, shared mission? Are we deploying our individual strengths and covering our weaknesses effectively? Have we all talked about how we each prefer to work and come to any necessary compromises to best accommodate all of us? If not, those are often starting points for moving the team to at least the beginning of a healthier place.

9.4 **Contributing to an inclusive workplace**

Another thing that you can do as a great team player is help make your workplace more inclusive.

Now, I want to pause for a moment and acknowledge that the topic of workplace diversity, in the tech industry in particular, is far too complex for a single section in a single chapter of a book; it's worthy of numerous books, all written by people of different perspectives. I'm in no way trying to suborn that effort. If anything, all I want to do here is highlight the need for those many, many books. I'm also completely aware that as a white American male, my own perspectives reflect only a portion of the vast spectrum of perspectives and experiences. Indeed, that is one of the big points of striving for more diverse workplaces: you end up with a more complete picture of that vast spectrum, as others' different experiences and perspectives help provide a broader and ever-more-complete view of the world in which we and our organizations exist.

That said, I simply want to offer a few suggestions to help anyone make small yet meaningful contributions as a part of everyday work life.

9.4.1 **Help**

Start by seeing whether you can do something to help widen the top of the funnel for technology teams. That is, even if today's workplaces aren't as diverse as they could be, what can you do to help make tomorrow's workplaces more diverse? Volunteering for organizations that help teach kids to code, build networks, or fix computers can be an investment in making future generations' workplaces better reflect the makeup of the world we live in:

- Black Girls Code (<https://www.blackgirlscode.com>) and AllStarCode (<https://www.allstarcod.org>) are two such organizations, focusing on Black girls and young Black men, respectively.
- I've also supported Tech Impact's IT Works program (<https://techimpact.org>), which targets kids from disadvantaged backgrounds and helps them get entry-level tech jobs.
- Latino STEM Alliance (LSA; <https://www.latinostem.org>) focuses on the Latino and Latina audiences, broadly covering science, technology, engineering, and math, including a wonderful robotics program.
- CODeLLA (<https://www.codella.org>) teaches Latina girls to code; there's also an informal Latina Girls Code movement with presences on Facebook and Twitter.

Those are just a few examples. **The point is to find these organizations and support them**, often with your time in helping teach your trade to young people who might not ordinarily have access to it. Maybe it has nothing to do with being a great team player today, but it has a lot to do with setting up future teams to be even better.

9.4.2 **Offer respect and support**

As you move through your workdays, be aware that we—every single one of us—have different backgrounds, come from different cultures, and were raised in different environments. Those differences make us valuable, but they also make us vulnerable. Every little thing that makes us stand out from the crowd is something that someone can use to push us down or make us feel ashamed. In my generation, a lot of us in the tech world went through that in school. I was one of two kids who helped out in our school's nascent computer lab, and I took all kinds of grief for it. *Geek* and *nerd* weren't fun terms back then, as they are now; they were a way of hurting someone. I was ostracized, and when I spoke up about it, I was told that I should "toughen up." That's a common phrase—"toughen up." When people complain about being made fun of, that phrase is used to make even more fun of them. And the people telling us to toughen up are usually the ones who aren't being made fun of at all and who are in positions where they'd never be made fun of.

Try to be aware of that fact. Nobody wants people making jokes about them without their consent. If that's never happened to you, you can't possibly understand how miserable it can make a person. Just waking up and knowing you have to go to work where people are going to make fun of you and who don't consider you their equal is **horrible**. Given how easy it is to not make those jokes, comments, or whatever else, how easy it is to say nothing or to offer a compliment? Why would anyone choose to be hurtful when choosing to be not hurtful is so much easier?

I'm definitely aware of the sentiment "Everyone has to be politically correct now; you can't say anything at all." I'm aware of the **sentiment** "Some people are snowflakes and need to toughen up." But you can still say plenty to people. You can choose to tell them that you hope they have a nice day. You can leave them room to speak in a meeting so that their voice can be heard. You can stand up for them when someone else is

being a **jerk**. If you're the type of person who likes to make jokes at the expense of other people, your life is not significantly diminished by holding your tongue or finding better jokes.

We all want to feel that we belong. It's a basic part of the oldest pieces of our brains: belonging to a tribe confers safety, which is one reason why humans are such social creatures. We all want to be part of the popular crowd in high school, and although most of us grow out of that as we grow up, we all still want to feel like we belong. You can do a lot on an everyday basis to make people feel like they belong.

I remember one job I had, which I got shortly after meeting the person I'm now married to. Most mornings at work, people would move in and out of the break rooms, getting their morning coffee and such, talking about the weekend or the previous evening. They'd talk about their wives and husbands or their kids. They'd invite me into their conversations—because, I thought, they didn't realize that I didn't have a place in those conversations—and I'd keep it neutral. "Oh, we watched such-and-such a show," I'd say, or "We went to a new restaurant"—always *we*, *us*, or something else neutral.

My co-workers weren't stupid, and they didn't miss my omissions. Finally, one of them, Mike, pulled me aside. "Hey," he said. "I notice you're always careful about how you talk about your family life. If it's because you have a boyfriend, just know that you're completely welcome to say that. We're all fine with it. It doesn't matter."

That one little invitation to belong made an enormous difference. I hadn't even realized that although I didn't dread going to work, it was an effort. It was an effort to edit everything I was about to say to make sure I didn't slip up. And now I didn't need to do that. That's the kind of little, low-cost things you can do every day: let your co-workers know that they don't have to edit themselves so long as they're being professional and respectful. (Believe me, I still edit the amount of swearing I do; if you come from a military community, you pick up certain words that aren't appropriate in a civilian workplace.) They're free to be themselves. They belong, exactly as they are.

Offering an inclusive workplace isn't something we should do as a human-resources exercise to prevent lawsuits. It should be something we do because we work with other actual human beings. We might not all recognize our sense of belonging at work, because we've never been without that sense. But if you think about what it'd feel like for someone else, even small steps toward alleviating that feeling for them is a true gift, and it's no less than they deserve as fellow human beings.

9.5 *Further reading*

- *You Are The Team: 6 Simple Ways Teammates Can Go From Good To Great*, by Michael G. Rogers (independently published, 2017)
- *The 17 Indisputable Laws of Teamwork: Embrace Them and Empower Your Team*, by John C. Maxwell (HarperCollins Leadership, 2013)
- *The Soul of a Team: A Modern-Day Fable for Winning Teamwork*, Tony Dungy (Tyndale Momentum, 2019)

- *Belonging At Work: Everyday Actions You Can Take to Cultivate an Inclusive Organization*, Rhodes Perry (PYP Academy Press, 2018)
- *Disconnected: How to Deliver Realness, Meaning, and Belonging at Work*, by Colleen McFarland (New Degree Press, 2020)

9.6 Action items

For this chapter, I want you to think about the kind of team you'd most want to be part of:

- Make a list of attributes that your ideal teammates would exhibit. How would they behave toward you every day? What would they bring to the table, and how would they interact with you in ideal circumstances?
- Referring to that list, which of those attributes do *you* exhibit toward your other teammates on a daily basis?
- Ask some of your teammates whether they're open to discussing the team itself. Ask them to create similar lists, perhaps anonymously. When everyone's made their lists, compare and contrast them as a group. Are there any common attributes that everyone would like to see? What differences exist? Those differences show how we're all a little different as humans and that we can't always just treat people they way *we'd* want to be treated: we need to treat them they way *they'd* want to be treated.
- Ask one of your teammates whether they've ever felt like they didn't belong at one of their jobs—whether that's the current one or not. Even if you've struggled with a sense of belonging, it's incredibly helpful to hear other people's takes on the subject and to hear about their experiences.

10

Be a team leader

A lot of technology professionals shy away from moving into management or leadership, and this is a shame. Although managing and leading certainly aren't for everyone, they can be a highlight of a career and give you an opportunity to have a positive impact on others' careers, provided that you're good at the job.

10.1 The decision to lead

A lot of my friends in the tech world have been clear with me: they have no interest in management. If I ask why not, they give a variety of reasons, but a few common themes emerge, often boiling down to this: they've had a poor experience with managers in their own lives, and they don't want to become that person.

I've had some terrible experiences with managers and some wonderful experiences with truly gifted leaders. At some point in my career, I looked back at how much I'd been helped along by the excellent leaders in my life and decided, "I want to do that too." I suddenly wanted to help make the decisions that guided the organization, and I wanted the opportunity to affect other people's careers in a positive way. I'm glad I came to that realization, because it's been incredibly rewarding.

Management and leadership aren't part of everyone's careers, and that's fine. However, that's not a decision you should make without reading this chapter. And if management or leadership is part of your path to success, I hope this chapter will provide some tips to make you happier and more effective at it.

10.2 Leadership vs. management

I want to nitpick some words here. The definitions I'm about to offer are by no means universal, but within this chapter, I will use them to distinguish between some different sets of responsibilities and behaviors.

Let's start with *supervisor* so we can put the term aside. In this chapter, a *supervisor* is someone who watches over a group of people to make sure they're doing what they're supposed to be doing. I understand that some people might not be attracted to that role, because they see it as not much more than babysitting in some regards. But supervisory roles can also be a way to dip your toe into management. You take on some greater responsibilities, and you start to explore what management and leadership mean in practice. If you've ever been on a team that had a team lead, such as a lead software engineer, you may have seen a great kind of supervisor in action, even if they weren't called that.

Next up my hierarchy is *manager*. Sometimes, a manager is much like a supervisor, making sure that people show up to work on time, disciplining people who break the rules, and so on. But really good managers are people who are given a set of resources and one or more outcomes to achieve, and who work to deploy those resources to accomplish those outcomes. A manager can be someone *on* a team, as opposed to someone at the head of a team. A lead software engineer working on a team of other software engineers might well have some managerial duties.

A *leader* is something entirely different, and it's what I'll focus on in this chapter. A leader is the person out in front. They're the one who shows everyone else what the destination is, outlines the path to success, and shows everyone the path that will lead them there. The leader is the one who helps everyone understand the mission. A great leader will help remove obstacles to ensure that the team can focus and get their jobs done. The leader is not the one who gets all the work done, and in a lot of cases, they might not even know everything that needs to be done. But they're the ones who create the vision for the team, and they're the ones who help everyone else understand how they can contribute to the mission.

Leaders are also the ones who take responsibility for the members of their team, treating those team members as people rather than resources. They worry about their team's state of mind, and they help guide their team's professional development. Leaders are the ones who build new leaders, helping people rise to whatever level their career requires. Leaders are the ones who recognize when someone on the team is no longer on the same journey as the rest of the team, and leaders are the ones who wish that person good luck when they head off to wherever their career takes them next.

Notably, a job title doesn't necessarily tell you whether someone is a leader. A manager of network operations can be a leader if they're exhibiting the characteristics of one. A "mere" supervisor can be a leader as well: it's all about how you operate, more so than the scope of your authority or your place in the org chart.

You may have worked for someone who called themselves a leader but didn't exhibit these behaviors. If so, they weren't what I consider a real leader. Don't let that

person's approach stop you from taking on a leadership role, **if becoming a leader is what will help your career move you to your success.**

10.3 **The leader's path**

Leadership is straightforward, in that there's a pretty short list of things you have to do. But leadership is hard, in that you have to show up every single day committed to that short list, and prepared to help your team recommit to the mission you're leading them on.

First, you create a vision—a description of where you see your team in the future. If you're leading an entire company, that might be a few years off; if you're leading a smaller team, it might be a year or less.

Your vision needs to set some specific and measurable goals that anyone with access to the right data could look at and say, "Yes, you hit it" or "No, you missed." But you're not looking at small, week-to-week goals; you're looking at the primary overall goals for the team—the big picture, in other words. These overall goals might include a description of the activities of the team itself, as well as specific achievements the team will accomplish. Your vision needs to be attainable too, and your estimation of what's attainable needs to come from your own experience, your research, and your data. If you can say, "Look, we got close to something similar in my last job before I left, and I know how it's done," that's a fine indicator of attainability. Or maybe your confidence in attainability comes from this thought: "Every other company who does what we do has achieved, or is achieving this, and I have all the data points to confirm it." That works too. But your vision can't be pure fantasy or gut instinct: it needs to be backed by some evidence to show it is a realistic, attainable vision. Let's examine a vision statement to see how to make it effective and how a leader uses it to lead.

Our team will become a diverse group of world-class Agile practitioners, and we will routinely deliver more, and more defect-free, builds across all of our projects.

That's a poor vision statement. It definitely implies that there are some goals and some areas where the team maybe isn't hitting what it needs to hit, but it does not provide enough precise, unambiguous information for a leader to use to guide the workings of a team. This statement is not objective, and it is not measurable. Exactly what is *world-class*, and how will we know if we get there? What does *more* mean, specifically? What about *diverse*—what does that mean in quantifiable terms? A true vision might look like this.

In four years, our team will consist entirely of certified Agile practitioners who work in defined, four- to six-week sprints. We will deliver 10 to 12 builds per year from those sprints, and we will maintain a defect rate of fewer than five blocking issues per build. At least 40% of our team will be from traditionally underrepresented groups.

That's a well-constructed vision statement. It provides a timeline of four years (and in reality, you'd probably update it every couple of years, always looking four years out, if that's the timeframe that works for you). The goals are presumably stretch goals, meaning that the team isn't currently exhibiting these milestones and isn't close to reaching them. The goals are measurable: the length of development sprints, the builds per year that result from those sprints, and a defect goal to try to beat.

This vision statement is a good one because you can start to look at where you are *now*, then look at where you need to be for the vision and start constructing a path between the two. As a leader, you might decide

- I need to have two members of the team become Agile-certified every quarter. I'll stagger them so that I can assign people time to prepare without taking the whole team offline.
- We currently work in one-year sprints, so I'm going to build our next sprint around a smaller set of deliverables and try to hit three months for that sprint.
- We're going to do monthly Agile reviews so that we can better reinforce Agile principles and make sure we're learning from our missteps.
- When new positions open, I'm going to commit to interviewing a diverse set of candidates, both internally and externally. I will always hire the best person for the job, but I recognize that bringing a new perspective to the team is one of the criteria for best.
- We will aggressively commit to unit testing as a way of life in an effort to reduce our per-build defect rate. I will need to put everyone through the right training, and we will monitor our improvement.

Those are actual, actionable, visible things that the team can do. As a leader, you can show that list to your team. Putting on your manager hat, you'll see that you need to juggle some resources, and you may need to do some negotiating up the org chart to get those resources. You might go to your boss and say, "You know how you said you wanted more, smaller builds that were more reliable? Well, this is what I'm going to need to get that done. We need to make some minor investments in the team and our processes, so you're going to get smaller builds as I carve off some of their time to do that."

Negotiating up the org chart

One important responsibility of any good leader is managing up. We often think of management solely as managing the people who report to you, but another behavior of a leader is managing the people they report to. I'll offer two examples.

My own boss came to me one day, explained some shifts in the company's priorities, and asked that I realign my team's activities to support those priorities. I looked at what was being asked and explained that I'd need to add at least one more software engineer to make it happen. "There's no extra head count," I was told. "Okay," I said, "but I need to take these two items that you've asked for and make them optional."

(continued)

We'll try to get them done, but I can't commit to it." I worked for a company that was supportive of that kind of negotiation, because the company recognized my greater ability—given that I was closer to the actual work—to understand what was possible.

In another instance, my team had lost a couple of people and not been able to fill those jobs, so we were short two team members, which put our outcomes in jeopardy. Rather than going to my own leader to explain the problem, I went back with three possible solutions. "I can't deliver what we originally agreed to," I explained, "but I've come up with three scenarios I feel we can achieve, given where we are right now." We used those solutions as the basis for a negotiation: my leader helped me understand the company's priorities at the time, and I helped her understand what the team could accomplish.

Negotiating up the org chart is a powerful skill, and it's one that a well-run company will understand to be a valuable part of the overall business.

That kind of up-the-org-chart statement is part of what makes a good leader. The organization tells you that it wants something, such as adopting an Agile approach or increasing productivity, and you, as a leader, have created a solid vision of it, as well as a path to get there. Nothing is free, and resources are finite, so you figure out what this vision will cost the organization in equipment, time, people, and money. As the leader, you are accountable for that investment creating the return that you've established, and you should welcome being held accountable for it.

With a good vision statement in place and a plan that takes you there, you start showing your team members where they fit. With a truly clear plan, it'll often be obvious where everyone fits. One team member might see that she needs to start studying for that Agile certification; another might say, "I guess I'd better start reading up on unit testing." From there, your job as a leader is keeping everyone on track. It's easy, in the day-to-day chaos of life, to lose track of the vision; your goal is to keep it in sight.

10.4 *Getting into their context*

A couple months along the path of putting the vision into action, a team member approaches you, saying, "We really need to add this one extra feature, because people have been calling the help desk about it a lot."

But the team is currently two weeks into the sprint. If you added that feature now, you would break your Agile principles and jeopardize the sprint itself. You refer to the vision statement to decide what to do. "Is adding this feature at this point going to help us achieve our vision?" And you decide that no, the vision is bigger than this one feature, and the team needs to stay committed to the vision. You explain your thinking to the team, but they're skeptical; they still think it would be best to add the feature right now.

What do you do if your team members disagree with you? How do you handle that disagreement, along with the many other disagreements, arguments, misunderstandings,

and fights that naturally crop up in the work environment? You handle it by getting into your team members' heads—by getting into their context.

A microprocessor can execute code in one of two spaces: kernel space and user space. Kernel space is a protected environment, designed for sensitive, operating system-level code that keeps the whole computer running. User space is for regular applications. The two spaces are separated so that a single app can't easily crash the entire computer.

The microprocessor cannot execute user space code and kernel space code at the same time. Doing so would allow one to affect the other, which would defeat the point of separating them, so the microprocessor must execute a context switch to change from one to the other. This action is a deliberate one that consumes time, and it lets the process shift its thinking. "Okay," it can say to itself, "I'm in kernel context now. I need to behave a little differently, and certain actions are going to be allowed that weren't allowed when I was in user context. Deep breath. Here we go."

As a leader, you need to become a little bit like a microprocessor and develop the ability to change contexts more or less on demand.

When you get done reading this paragraph, I'd like you to sit back for a minute and close your eyes. Float around in your own mind, and observe your internal monologue. What do you think about when you're not thinking about anything? I just did and found myself thinking about a friend who's driving up to visit this weekend. We've not seen him for a while, and I'm looking forward to hauling out a nice bottle of wine and catching up. I'm also worried about the raccoons that have been knocking things over on the deck every night and what we're going to have to do to put an end to that. Oh, I need to take the trash out too. I could use a glass of iced tea. My jaw kind of hurts, I must have slept on my face wrong or something. Or I've been clenching my jaw again. Ugh, I haven't been to the dentist in like nine months.

That was my internal monologue for a moment of the day. That was my context. It's where I was in my head. It's why, when my beloved spouse just asked me whether I'd checked on the chimney sweep, I snapped, "No!" I've already got a lot of house stuff on my mind, and this is just one more thing. Combined with the pain in my jaw, it wasn't what I needed to hear or how I needed to hear it.

Then we had a fight, of course, because I snapped. The thing is, my darling spouse is in another context altogether different from mine. My in-laws have been going through some difficult financial decisions, and I know the situation is affecting my spouse and their siblings, creating a lot of worry. My spouse's context was trying to make sure that this chimney sweep got taken care of, because so much else is going on with the family that it'd be easy to let it drop. And here I am, snapping for no reason. Well, not for no reason. In my head, what I did made sense. In my spouse's head, what they did made sense.

We're all operating in different contexts. We've got different thoughts that are top of mind, different concerns that are worrying us, and different perspectives on what's going on in any situation. Those differences are what create strife. As a leader, it's your

job to put yourself into other people's context so you can understand what's happening. It is not their job to suck it up and get the job done; it is your job to lead. To lead, you need to know a bit about the people you expect to follow you. People aren't going to follow you if they think you're headed someplace stupid; you need to understand what they think is smart and stupid, and help them understand, in their terms, why this direction is the smart way to go.

Team members: "Agile is stupid. You can't get anything done in six weeks."

Team leader: "Well, smart or stupid, what the company needs from us is to be more responsive. We don't have to use brand-name Agile if we don't want to, but I do want to look at the principles and start shifting toward more, smaller releases."

Team members: "That's just another dumb change. We do this every two years—change our approach for no good reason."

Team leader: "Hey, there's obviously something here you're worried about. Tell me what you're thinking."

Team members: "It won't matter. You'll just do what you want."

Team leader: "Maybe I won't. Forget what I think about this for a minute. If I'm wrong, I'm wrong, and I'll admit it. Tell me what you're thinking. You're definitely a little upset, so tell me why. If my plan is dumb, I don't want to do it. Just tell me what's up with you right now."

To get the context of someone else, you need to ask questions. I understand that being touchy-feely isn't comfortable for everyone. It's not comfortable for me. I'm a sarcastic, engineer-brained computer nerd who'd be more comfortable assembling a server than talking about feelings. But when I'm in a leadership role, it's my job to understand where my team is, mentally and emotionally. If I'm going to convince them to follow me, I need to understand what matters to them, what their priorities are, and what they care about. I need to find a way to align them to what I need us to do together, and to do that, I must ask them a lot of questions about themselves. It's those questions—and answers—that will get me into their context. You know what? Maybe I'm wrong and they're right, and they're just not good at expressing themselves in a way I understand. It's my job to shift so that I can understand their perspective.

So why can't you expect your team to suck it up and do their job? They're people. People don't suck things up. They may try to ignore their feelings and act like

everything is fine. But when you encourage or allow them to do this, you're not creating a healthy team. You're not creating a team that will follow you.

Being a true leader is not a right that you're handed along with a new job title and a pay raise; it's a privilege that you have to earn. Remember that as the leader, you're probably the least able to get things done of anyone on your team. Your team members are doing the work. Your job is to earn their followership and help them get to the outcomes everyone wants.

I acknowledge that not everyone is going to be the perfect team member. If you present your vision, and some team members think the entire vision is wrong, they might not be a great fit for your team. "This is where we're going," you might say, "and you obviously aren't required to come along. But this is where we're going, and if you don't want to come, we need to work on finding you a team that's going in a direction you'll be interested in."

This isn't a "My way or the highway" approach: if the vision you've laid out and the path you've created are the ones that will achieve what the company needs to achieve, the team eventually needs to be on board with that. You would certainly ask your team for feedback, and you might ask for their help in designing the right path to the vision. If you're truly collaborating and helping them understand the outcomes, and why those outcomes matter to the company, a good team will rally with you and work to get the job done. But there will be times—although I hope they're few—when a current team member isn't interested in the new journey. It's your job to help them find a journey they *do* want to be on.

10.5 **Leading positively**

I've worked for more than a few so-called leaders who try too hard to make themselves one of the pack. Any time they're asked to do something they know will be unpopular, they try to commiserate and put the blame elsewhere.

"Look, upper management says we're going to have to put in whatever hours it takes to get the server migrated. I know, I know—I told them we'd been trying for a week already, but they're just not listening to me."

That's not leading, and it's not managing. I'm not sure what it is except a little annoying. Remember: leaders have a vision. That vision creates outcomes that the organization needs. Leaders develop a path to that vision, and they work to inspire their teams to stay on that path. They help remove obstacles between their team and the vision. And leaders take accountability.

Whenever you, as a leader, start from a place of negativity, you sabotage your own path to your own vision. When you're negative, the entire team is entitled to be negative.

That will reduce their productivity, affect their own mental health and happiness, and lower their desire to follow your lead.

"Folks, it's come down to it. We all know sales needs this server migrated before the new campaign launches, and that's now next week. We've spent a week fighting this thing, and you're telling me that we've got the last hurdles out of the way. Tomorrow's the drop-dead date we gave them, so we need to make this happen. I know that's going to mean us working all night, and I'll be here with you, but this is what we said we'd get done, and we need to get it done. Other people are relying on us. From what I'm seeing, we don't need all six of you here all night. Do you want to break this down into shifts?"

Leading positively doesn't mean you have to put a smiley face on every situation. Bad news will come, and bad times will occur. Leading positively means being able to say, "I know this isn't great, but we need to get through it. How can we do that?" It's looking forward to the vision. It's reminding everyone of the role they play in bringing the vision to life. And it's letting them know that you'll pull your weight, just as you are asking them to do. If you've created an attainable vision and outlined a clear path to get there, all that's left is to do the work.

10.6 **Mistakes leaders make**

We all make mistakes. The point of having experiences is to learn from them. And the point of teaching, as I'm trying to do with this book, is to share your experiences so others can try to avoid the mistakes you've made. With that in mind, here are some mistakes that leaders can make:

- **Being dishonest**—If your organization is doing something stupid, you don't have to paint a pretty face on it. But you also don't need to commiserate about it with your team members. Refocus your team on the vision. Rework the path to that vision, if you need to. Reshow everyone what their role is.
- **Blaming others**—Leaders are accountable, period. Yes, your team members might make mistakes, and in some unfortunate cases, you may even have to take formal disciplinary action in response to those mistakes. But make sure you're looking at ways to keep those mistakes from happening. Micromanaging isn't the answer; making sure that everyone knows their role and has objective metrics with which they can determine their success is the way to minimize those mistakes.
- **Stifling ideas**—Most technology professionals are passionate, creative people who want to improve the world they're in. Don't stifle them. Let people express their ideas, and listen to those ideas. Accept that your team members might be a lot smarter than you, and rely on them to help you construct the path to the vision.

- **Letting mistakes go**—Mistakes provide an opportunity to learn. To make the most of mistakes, discuss them as a team, and decide on a new way forward to prevent that mistake in the future. It's also important to document mistakes in project notes and records. Create an environment of fearlessness in which people are comfortable being accountable because they know failure will be treated as a learning experience, not followed up with punishment.
- **Talking too much**—“Talk less, smile more,” as the *Hamilton* lyric goes. As a leader, your job is to help your team see the vision and then support them in getting there. You can't support people unless you're listening to their ideas and their needs.
- **Communicating poorly**—Your written and verbal communication skills are critical to your success as a leader. Remember that not everybody reads an emoticon :P the same way, for example. Work to communicate in as unambiguous a way as possible, and be aware that your communications may be received in a way other than you intended. Listen, learn, and adjust.
- **Forgetting**—You've had bad leaders in your life. Don't forget them, and don't be them.

10.7 Leadership beyond leading

One final task of a truly effective leader is taking care of your people. Although this task obviously includes protecting them, when you can, from upstream static and helping them stay focused on the mission, it also means trying to take care of your team, both as people and as professionals.

Take an interest in your team's own individual definitions of success. Ask them what kind of life they're hoping to lead and what they've defined as the kind of career that will get it for them. Ask how you can help create that kind of career for them—by offering advice, if nothing else. When you see something in their career definition that you know you can help with, offer to do so. Maybe you can help by coaching them to become leaders themselves, or maybe you can help them understand more about the company so that they can better align their current job with their career goals.

I've been fortunate in that I've almost always had leaders who wanted me to grow into whatever kind of role I wanted. One of my first jobs was working part-time as a retail sales associate for a computer software retailer (Game 'n' Gadgets, if you've ever heard of them). Most retail outlets have a manager and an assistant manager, and often one or more keyholders. Those keyholders have a key to the shop and are permitted to open and close the store, including counting down the till and doing opening or closing paperwork. My manager didn't care for the keyholder concept: in her eyes, everyone needed to know how to do everything. And although I worked a scant 12 hours a week there, I quickly learned everything. I could do most of her job, and when an opening came for an assistant manager at a nearby store, I got hired easily. I've never forgotten her attitude.

Even today, I make sure that everyone on my team knows how to do my job, even if they don't have the authority—yet—to do it all. One day, there'll be an opportunity for them to move up, and it'll be an easy decision for them and for the company. All this depends on whether they want to move up in this direction, of course. I've been careful to ask about their own goals and values so I don't push them in the wrong direction. In some cases, I've suggested that they focus on a particular aspect of their job for growth, and we've discussed how that could help move them into a job that's more closely aligned with their goals than the job they currently have. I hope that the new job will still be with my company, but if it isn't, I'm happy that they're doing what's right for them.

10.8 **Before moving into leadership**

As I've spent much of this chapter defining leadership and outlining what it can look like, I want to take a moment to argue against a move into leadership. I moved into my first leadership role well before I was ready and wound up leaving the company to move back to an individual contributor role. It was more than a decade later that I finally—hesitantly—agreed to move back into leadership, and I found myself much better prepared. While writing this book, I had some colleagues—one from my own team—go through the similar step-into-leadership-and-step-right-back-out experience, so I think it's well worth questioning a leadership opportunity if it comes your way. Do you know what the day-to-day work in this leadership position entails? Is that the kind of work you like? Do you have all the skills and experience you need to do well in that position?

Never accept a leadership position solely for the title or the money. Management is often seen as the only way up, and in many organizations, that's true in terms of salary. But moving solely for salary can backfire. Instead, move into leadership because you want to lead and because you want to excel at it. Leadership is a skill like any skill, and although it's a skill that I think anyone can master, it's a type of work that not everyone will enjoy. That work differs across organizations as well, so make sure to gather information and analyze what you're getting into.

10.8.1 **Don't get promoted to your level of incompetence**

In the classic book *The Peter Principle*, published in 1968, Laurence J. Peter proposed a concept that's relative to our discussion of whether to move into leadership. Broadly, it states that people in a hierarchical organization tend to get promoted to their “level of incompetence.”

It goes something like this. Let's say you're a great entry-level software developer. You do well, and you eventually get promoted into a midrange developer role. You struggle a bit at first with the additional job duties, but then you master them and start to succeed. After a time, you end up in a senior developer role. This role adds some team-coaching responsibilities, and again, you struggle a bit at first, but eventually, you find your groove and start to do well. Next, you're promoted into a full leadership role

as a software development manager. You're no longer working on code and are instead responsible for leading entire teams of your former peers. The pay is great, and it feels good to have the company's trust. You struggle a bit at first, because it's an all-new kind of work, but eventually . . .

You keep struggling, and struggling, and it never gets better because you've been promoted to your level of incompetence. Because you're not succeeding, it's unlikely you'll go farther. It's not even your fault, really: you've just been put into a job for which you don't have the right skills. Your amazing abilities as a developer did not translate to amazing leadership skills. Another way of looking at it is that you landed a job you're not qualified for. It'd be like a UNIX systems administrator somehow landing a job as a data analytics architect; the pay would be great, and the job might seem interesting, but your old skills didn't in any way line you up to be successful.

Try to avoid getting promoted to your level of incompetence. If you want to be promoted into leadership positions, learn what those positions entail and what skills you need to do well at them.

10.8.2 Learn leadership

Leadership skills *can* be learned. The process is similar to learning a new technical skill, right? You learn a bit, you practice a bit, and you take on small roles to prepare yourself for larger ones. Here are a few ways to learn and practice leadership skills so you can ease into a leadership role when and if the time is right:

- *Establish a mentoring relationship with an existing leader in your organization.* I enthusiastically recommend this as a way to find out what the job is really like, including all the ugly parts (such as having to fire people or lay them off, or having uncomfortable discussions about job performance, for example).
- *Take on a tech-lead position if your organization offers this option.* In a tech-lead role, a skilled technology practitioner informally picks up some leadership-assistance duties, often in exchange for a small salary supplement. This is a way for the organization to spread out the leadership workload somewhat, and it's a way for someone to have a trial run at a leadership-light position.
- *Volunteer for organizations with leadership opportunities.* Ideally, you'd start on a small scale, leading a team or a project, to gain experience and see whether you like being a leader.

10.8.3 Measure your own success

As an individual contributor, it's often easy to measure the success of your days. Perhaps you got a server up and running, finished a module of code, got a data visualization working, or something like that. You produced something, in other words. As a leader, it can be much harder to tell whether your day went well. You'll often sit in more meetings, for example, and although you might be conveying marching orders to your team, they're the ones getting the marching done, not you. So leaders have to be able to find new ways of deciding whether they had a good day at work

or not. If you aspire to be a leader, you have to be comfortable with those ways, if you find them.

As a leader, I've tried to shift my focus to "Did I have a good week?" instead of evaluating day by day. I do this because sometimes my job does not result in a daily win. So I make a special effort to write down things that went well, as well as things I feel that I could have improved. At the end of the week, I review that list, and I usually save my lists for several months. This is sometimes the only way to remind myself, amid the chaos of everyday management life, that I did get a win in here and there.

10.9 Further reading

- *Brave Leadership: Unleash Your Most Confident, Powerful, and Authentic Self to Get the Results You Need*, by Kimberly Davis (Greenleaf Book Group Press, 2018)
- *The World's Most Powerful Leadership Principle: How to Become a Servant Leader*, by James C. Hunter (WaterBrook, 2004)
- *Powerful Leadership Through Coaching: Principles, Practices, and Tools for Leaders and Managers at Every Level*, by Michael K. Simpson (Wiley, 2019)

10.10 Action items

For this chapter, I'd like you to take some time to think about leaders you've found effective and enjoyed working for and consider some of the attributes they exhibit:

- Which leaders have communicated a vision to you that you understand and that you can see your place in? How did they do that?
- How did your best leaders inspire you to help make their vision come to life?
- How have the leaders you've worked with tried to understand your context so that they could better help you understand what they needed from you?

11

Solve problems

Problems are the spice of life at work! They're the situations that take us out of the day-to-day grind and give us a challenge to work on. If you're not a confident problem-solver, of course, problems can be stressful and feel like opportunities for failure. But with the right methodology, you can turn problem-solving into a repeatable, reliable process that makes you a winner.

11.1 Problem-solving vs. troubleshooting

I want to emphasize that this chapter is about *problem-solving*, not *troubleshooting*. I do realize those terms seem to be awfully alike, though, so I want to take a moment to differentiate:

Troubleshooting, in my mind, is a technical activity. It happens when something is not working properly and you are seeking to restore proper operation. Troubleshooting has a fixed and definite outcome: restoring whatever is broken to its proper functionality.

Problem-solving, for me, is not necessarily a technical activity. It can happen between people and between businesses, and it doesn't necessarily indicate that something is broken—merely that something isn't working the way you or the other party most want it to. The outcome of problem-solving is often subjective; there isn't always a fixed solution to aim for, but a spectrum of possible solutions, each with various trade-offs.

11.2 Clearly state the problem

To illustrate how I think about problem solving, I'll use a running example in this chapter, along with some simple diagrams. Let's say I have a vendor, whom I've

contracted to create a particular deliverable for my company. Maybe that deliverable is an application of some kind, a document, or an analysis of some sort. Doesn't matter; it's a deliverable. We've agreed to a price and to a delivery date. In other words, we've got full alignment between what I want and what the vendor is delivering. Let's express that agreement in a diagram. In figure 11.1, I'm the circle, the vendor is the square, and we have direct alignment between us.



Figure 11.1 Direct alignment = no problem

Now let's throw a problem into the mix. In figure 11.2, something has changed. Let's suppose that some internal business pressures led me to need the vendor's deliverable earlier. Now the vendor and I are no longer aligned: the vendor was prepared to do what they had contracted to do, and I'm changing the rules of the game all of a sudden.

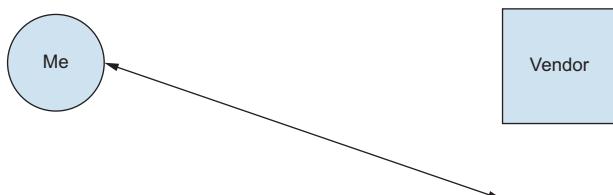


Figure 11.2 Misalignment = problem

As the person who needs to solve this problem, I'm wondering how to move the arrow back into alignment. If you want, think of this figure as a billiards illustration, with the arrow representing the path of the cue ball. What else can I add to the diagram so that I can bank the cue ball off something and get it into the right pocket: the vendor?

Avoid overfocusing on the source of misalignment

You'll notice in this scenario that I haven't said why the company's timeline for delivery has changed. When I'm problem-solving, I try not to focus too much on the cause of the problem; instead, I try to focus on the nature of the misalignment and finding a solution to put into place. This attitude is different from the one you'd use when troubleshooting a technology problem. In troubleshooting, we often need to examine the problem clearly because it gives us clues to the solution.

But in problem-solving, it doesn't do a lot of good to spend much time thinking about why my company's delivery timeline changed. Maybe the company is getting pressure from a customer, an executive is acting on a whim, or something else. Most often, I can't affect whatever is driving the misalignment, so focusing on it wastes time and makes me unnecessarily frustrated.

11.3 Identify your levers

I start by making a list of levers—things about the situation I can change. Not all the levers have to be helpful; at this point, I’m just trying to figure out what I can influence in the situation. In this scenario I might consider levers such as these:

- *Money*—Perhaps the vendor could be motivated by additional money on the table, although my finance department might not be able to give me more money.
- *Relationship*—Presuming that the vendor is motivated by having work to do, they might find value in having a better relationship with my business—a relationship that could result in more work over the long term. So if I can guarantee more work, the vendor may be able to deliver on the new timeline.
- *Deliverable*—Can I change the nature of the deliverable so that I’m asking the vendor to do less, thereby making my new timeframe easier to achieve? Could I remove some application features, reduce the length of a document, or make some other substantive change?
- *Alternatives*—Could I switch vendors? This lever can be a complex one, depending on the contract, the business relationship, and other factors, but it’s something I can consider.

Suppose that I go to the vendor and say, “Hey, I can throw in 10% more money if you can get this done faster.” But that lever, shown in figure 11.3, might not be enough. The vendor might come back and say, “Well, I can get you more, but I still can’t get you the whole thing.” So I’m still not fully aligned.

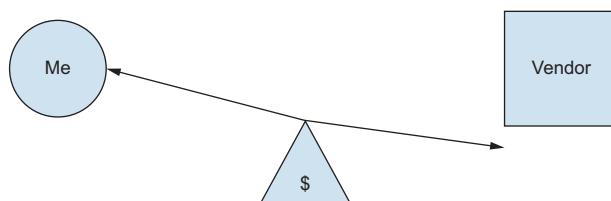


Figure 11.3 Using levers to realign

That lever helped, but it didn’t solve the problem. Okay, let’s try another lever. I’ll reduce the deliverable as well and try to pull the line up—only a little, perhaps, but enough to get us to an agreement. Figure 11.4 shows the final solution.

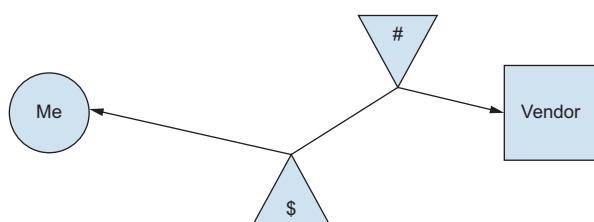


Figure 11.4 Using multiple levers to create full alignment

This example is the essence of problem-solving: clearly state the problem and then clearly identify the levers that you might be able to use to influence the situation. Pull the right levers to re-create alignment and remove the problem.

When you approach problem-solving this way, the tough part is knowing what levers you have for any given situation. That's where experience comes in, and a deliberate effort on your part to learn can play a big role. Knowing which levers apply to a particular situation is something a mentor or manager can and should be able to help you with. Also, you can perform independent research—within your organization, by talking to your co-workers, and by researching your field in general—to broaden your understanding of the situation and the levers you can pull.

Problem-solving and managing up

On more than one occasion, someone who worked for me came to me with a problem but didn't bring any possible solutions to me. They were asking me to solve the problem, in other words. In those situations, I always push back.

You hear the phrase *managing up* a lot, and I find that problem-solving is a perfect situation in which to manage up. What does that mean?

Rather than asking me, the leader, to manage down and solve problems for my teams, I want my teams to bring me potential solutions along with their problems. I want them to understand which levers apply to the situation—something I'm always trying to help them do—and bring me a small selection of recommendations. For each proposed solution, I'd like to know which levers they're suggesting we pull and what outcomes they expect to create.

Managing up is bringing solutions to your leader rather than expecting them to provide you solutions.

One option might be to spend more money and reduce the feature set a little for the initial release. Another option might be to stop development on another project and move those people over to this project.

Every situation has downsides: pulling a lever always means that you're incurring a cost or giving something up to get something else. Having multiple solutions and being able to quickly understand the trade-offs each one represents help me make better decisions in the end.

Managing up also creates a promotion path for the people on my team. Because they're exposed to the decision-making process, and because they learn which levers are available to pull in a given situation, they start to do some of my job for me. Being able to do my job is how they ultimately get my job, whether that's by way of a promotion within the company or by taking their new experience and getting a better role somewhere else.

11.4 Negotiating solutions

Negotiating is a major aspect of problem-solving. I find that people sometimes miss two major components of negotiating, so I want to call them out:

- *In any healthy negotiation, both sides usually gain something and lose something.* I might pay a vendor more to deliver something faster, but the vendor is going to have to work harder to make it happen. That's why some of the best negotiations end in a situation in which nobody may win, but nobody loses.
- *See negotiating as a way to try to achieve the best possible balance between opposing interests.* Negotiating isn't about winning and losing, and you should try not to see it as a competition. Negotiating should be about working together toward an outcome that meets everyone's needs in the best way possible while acknowledging that neither side will get everything exactly the way they want.

Why car buying is a terrible example of negotiating

I don't know how the process works outside the United States, but here, a lot of people consider buying a car to be the ultimate test of negotiating skills. Some of my friends get a real thrill out of "negotiating" \$1,000 off the price of a car.

Here's the thing: car-buying isn't really a negotiation. You want to buy a car, and the dealer wants to sell you a car. Therefore, you have full alignment in the beginning. But if the dealer's starting price is too high for you, what levers do you have to pull? Offering to pay cash isn't really an incentive, as dealers make a lot of money by initiating loans; in fact, customers who pay cash are less valuable.

You ask for \$1,000 off the price, but in exchange for what? What does the dealer get in exchange? Your undying gratitude? You've got nothing tangible to offer in exchange, meaning that buying a car is not a negotiation.

So what is it? It's a game. The dealer has an incentive to guess when you're likely to walk out and is fully aware that you're looking for the thrill of a win. So dealers and car manufacturers have all kinds of tactics to make you think you're winning while they get exactly what they want. They give you your \$1,000 off, and you're happy, but they never tell you that the manufacturer has a standard back-end \$1,500 rebate to absorb that cost. Or they front-load the price of the vehicle by adding paint-protection packages, nicer floor mats, and other details, all of which are deliberately overpriced to give you room to "negotiate" down.

Again, in any healthy, true negotiation, you lose something every time you gain something. Any time that doesn't seem to be happening, you're not negotiating; you're playing a game.

Negotiating, then, is all about understanding which levers you have to pull and how much you lose and gain by pulling each one. Here are some short examples that illustrate good and bad negotiating techniques:

- You're negotiating a base salary at a new job. The employer offers \$150,000, and you ask for \$10,000 more, but you offer nothing in return.

This example is a *poor* negotiation: you're trying to pull on a lever for money, but you have nothing on your side of that lever.

- You're negotiating a base salary at a new job. The employer offers \$150,000, and you ask for \$10,000 more; then you point out that the median base pay in your region is \$8,000 higher than the employer offered and that the number of applicants in the job market is extremely low, but the number of open positions is quite high. An extra \$10,000 puts the employer a bit over median, but it's an acknowledgement that the market is slightly against the employer right now.

This example is a *good* negotiation: you've brought data to the table, and you've put something on your side of the lever: the condition of the market and its ability to provide qualified candidates for all the open jobs.

- Your boss asks you to pitch in on a particular project to help get it done by the end of the week. You offer to drop some of your other duties to help out but are told that all those tasks have to be done as well. You object, pointing out that you have little enough free time as it is, and your boss counters by noting that you're well paid and that few other companies would hire you at that rate right now.

I think the boss's attitude is pretty toxic, but at least the negotiation is *valid*: the levers of time and employment are seeking to create a balance. Mind you, threatening a staff member's employment is not a lever that any boss should pull lightly, because it's going to encourage the employee to pull back by seeking a new job. There's a lesson here: just because a particular lever may be valid doesn't mean it's wise to pull it.

- You're having difficulty with a vendor that delivers great work but almost always misses their original agreed-upon deadline. You want to keep working with them, so you offer them a series of contracts. Your caveat is that they have to hit every deadline to guarantee that long stream of work; if they miss one, all the future contracts are void, and you're free to find a new vendor.

This example is a *good* negotiation: you're pulling a lever (offering a long-term stream of work) in exchange for something you want.

As you move any lever, it moves away from you and toward something else or toward you and away from something else. In perfect business relationships, all the levers in a situation are carefully balanced to create the best set of trade-offs.

11.5 Action items

For this chapter, I'd like you to consider a few experiences in your own life and workplace that involve problem-solving. There are no right or wrong answers; the exercise is more about thinking through the situation and becoming more automatic about problem-solving science:

- Suppose that you're not happy with the price that your internet provider charges for service at home. What levers do you have that would apply to this situation?
- Think about a few problems you've experienced or observed at work in recent months—internal problems, or team- or company-level problems that may not have affected you personally or directly. Where was the misalignment that created the problem? Which levers were applicable to each situation? In the end, which levers were pulled, and to what effect?
- What kinds of unhealthy negotiations have you witnessed recently, at work or in your personal life? Unhealthy negotiations are those that involve only give *or* take rather than the give *and* take of healthier negotiations.

10 *Conquer written communications*

There aren't many things as important to a successful career as our ability to communicate effectively with our fellow human beings. But effective communication requires a level of thoughtfulness and practice that many of us lack. Fortunately, you can take some straightforward steps to make your own communications more professional and compelling.

I want to start with a crucial point: this chapter is about *communicating*. Writing and speaking are two media through which communication can take place, but in this chapter, I'm going to focus on written communication. Nearly everything you learn in this chapter will apply to chapter 13, which is about verbal communication; I just find writing to be a little easier to conquer, so I'm starting with that topic.

12.1 *Communicating is telling a story*

The purpose of communicating is to convey information from one person to one or more other people. If people were computers—infinitely patient, absolutely unselfish, with perfect attention spans and absolutely perfect recall—communicating would be easier. But people aren't computers: they have their own priorities, they lose interest in things, and they don't have perfect memories. That's why being able to communicate well requires you to do more than just *spew information* at someone: you need to package your information in a way that will best achieve the outcome you're hoping for.

A co-worker at a former company once raised an issue in a team meeting:

I wanted to bring up a point, because I think it's an important point, and I was just wondering if anyone else felt this way. During that last reorganization, we had said that we wanted to create teams that were aligned to customer outcomes, but that we knew there would be certain teams that had to be in the background and provide support for shared service. Now, that's fine, because obviously it's more efficient, when you look at things like payment processing or authentication or something like that. Those things are all shared services, so it makes more sense to have them centralized into one place. But I was talking to Marcy about it, and you know, some of those shared teams really do have customer-facing outcomes. Because payment processing is something that does surface a user experience and obviously things like security of credit card information is important to customers. So those background teams aren't always really in the background if you think about it that way. Does that make sense?

Um, no. No, it does not make sense. You've rambled for five minutes, and I have no idea what point you were trying to make. That's because you didn't tell a story.

12.1.1 The rules of storytelling

Forget about communicating for a minute, and think about some of the best short stories you've ever read. Fairy tales are good examples, if you can't think of any other examples right now. All these stories tend to follow a set of rules:

- They have a clearly defined hero (or several heroes) at the center of the story. The hero's perspective is what we follow, and the hero is the one we're meant to root for.
- The hero always encounters a problem of some kind.
- The main point of the story is following along as the hero solves the problem.
- Particularly in short stories like fairy tales, there are **no tangents**, side stories, or other distractions. We stay on the main storyline with the hero.
- Things in the story happen to the hero and other characters, not to the storyteller. The storyteller is there to relate the story, not to be part of it.

What were Hansel and Gretel's parents having for lunch that day? No idea. What financial difficulties was Cinderella's stepmother struggling with? Not a clue. What kind of arguments were the Seven Dwarfs having among themselves? Doesn't matter. Fairy tales and other well-written short stories are notable for staying on track.

In any communication, whether it's a quick direct message in Slack, an email to your team, or a contribution in a meeting, you need to tell a story, and to get good at it, you can practice following the rules of good storytelling. Telling a story can be especially difficult when you're speaking off the cuff, but practice will make it easier over time.

START WITH WRITTEN COMMUNICATIONS I've been presenting at tech conferences since the late 1990s, and I'm pretty good at creating an ad hoc story when I need to. Yet I still tend to lean on written communications more. If there's a meeting coming up, and I have a point to make in it, I'll send a pre-read document, because writing gives me more time to consider, reflect, and edit the story I want to tell. Even when I plan to present verbally, I'll often do something in writing ahead of time because it helps me frame my thoughts, iron out my narrative, and make sure I'm delivering the right message. Writing can be edited, but speaking can't!

12.1.2 Applying storytelling to business communication

Now let's take those rules of storytelling and apply them to everyday business communications:

- When you're communicating with someone, make them the hero of your story. Making the story about them forces you to switch to their point of view for at least a moment and to empathize with their problems. The best way to get someone to engage in what you're discussing is to make them part of it. When you are speaking to a group that you belong to, it can also be effective to make *we* or *us* the focus of the story to emphasize that "We are all in this together."
- Clearly empathize with your hero's problem. State this empathy concisely; don't wrap a lot of unnecessary words around it.
- Keep your story focused on the hero and their problem. Don't bring in a single piece of information that doesn't relate to the hero, the problem, and the problem's eventual solution.
- Introduce no tangents. Stick with your story. It's fine to provide historical context to help people understand what you're communicating, but limit it to the absolute minimum necessary.
- Keep yourself out of the story as much as possible. Remember that nobody's in business for you. If you're trying to win people over to your cause, line up that cause with the team, the department, or the company. In business, those groups are usually the heroes of the story.

Here's an example of telling a story in a memo that puts these principles into practice:

When we created this new org chart, the company wanted to ensure that each team was focused on a customer outcome. The intent was to give everyone line of sight to the customer and their needs so that we don't lose track of that. That has worked well for those teams, but it has left out the teams that are considered background or shared services, such as payment processing. As a result, those background teams do lose track of what's important to customers, as we saw with the recent data breach. I would argue that every team actually does contribute to a customer-facing outcome and that we should rethink how we manage and motivate those teams.

Let's deconstruct that story to see what's happening under the hood. **The hero of this story is *we*, meaning the department or the company.** The story needs a tiny bit of background to set the context. That context comes at the beginning of the story, which makes sense chronologically, and gives the story a forward progression (so that it doesn't go back and forth between then and now). The storyteller presents a problem that affects the hero—our team or company—and briefly notes a relatable example of that problem (the data breach). The empathy comes from commenting on who has benefitted and who has not. The storyteller doesn't propose a full solution—often, you can't solve the problem, so that's fine—but they do suggest a next step on the hero's journey. This story is largely devoid of unnecessary words, tangents, or unrelated information. It is concise, and the storyteller's point can be understood by anyone who's listening.

Even the shortest direct message can be made better through concise storytelling. Consider this rather blunt directive:

Dave, I think your team needs to refactor this whole module of code.

Now consider a slightly longer, more story-driven version:

Dave, I've got to submit another request against that module your team owns. I know that module has gotten huge, and I know you guys are drowning in requests. Would it make sense to talk about refactoring that into four or five modules so we can spread the load out a bit?

Dave's the hero. You've acknowledged Dave's problem in a way that he can likely recognize and relate to. You've offered a first step toward a solution. You've not demanded a solution, but you've offered to be a part of one in a way that still involves the hero. The revised communication didn't take much more time to write than the first one, and it's more likely to generate the response you're hoping for.

12.1.3 **What about mundane, everyday communication?**

Not every communication needs to **persuade** someone to do something. Sometimes, you're just delivering a status report. Consider this status update:

The build server is down.

Is that an example of pro-level communications? It might be. It depends on your audience.

This missive does, after all, put the audience in the hero's role. If the people reading that communication will understand it and know what to do with it, you've done your job: you've acknowledged the audience's role as hero and given them the most concise story possible. This update would be appropriate and effective on a tech department's Slack channel, for example. Everyone who reads it knows what happens when the build server goes down, how it affects them, and what they need to do to prepare for it.

On the other hand, if your audience is wider than the tech department, and if the readers are left wondering "When will it be back up? Did I break it? Will I be able to work while it's down?", the writer did not do a good job of telling the story. They didn't acknowledge the audience's role as hero in the story or see the problem from the readers' perspective. Here's an alternative:

The build server is down. The cause is under investigation, and we hope to have it online in an hour. In the meantime, you can continue checking in code, and builds will resume when the server is online.

For an audience of developers, this communication might be more effective. The keys are knowing your audience; understanding how the status will affect that audience; and acknowledging that they, not you, are the important ones in whatever story you're telling, no matter how short that story is.

12.2 **Facing our fear of communicating**

When people have trouble communicating well, the difficulty usually **stems** from

- **Lack of practice in focusing on what makes for effective and ineffective communications**
- **Fear**

In this section, **we'll deal with the fear part; the rest of this chapter will address deliberate focus on being effective.**

As I wrote in the first part of this book, **fear is a powerful human motivator**. We often go farther to avoid a **frightening situation** than we do to achieve something we want badly.

For most people, the fear involved in communicating is a fear of embarrassing ourselves: we're afraid of looking stupid, and nobody wants to look stupid. Standing up in front of a group of people, we fear that we might ramble, forget the point we're trying to make, or speak too softly. We fear that this failure to communicate well will cause people to lose respect for us, which might affect our ability to move up in our career. Worse, we fear that people will mock us. Even worse, they might mock us behind our backs. So much can go wrong that maybe it's easier to stay quiet. Imposter Syndrome plays a role as well, because Imposter Syndrome is essentially rooted in our fear of being found out that we're not smart enough to deserve to be in the room.

The fear of looking stupid often hinders people from speaking up in meetings or giving presentations, but it can affect written communication as well: writing reports or important email can be nerve-wracking for similar reasons. In fact, one of the reasons I used to reply to emails with extremely terse answers was that I was afraid to write longer, more meaningful, and more compelling responses!

Let's be clear: if you're afraid to communicate, you're locking a big ball and chain on the leg of your career. No matter what you define as your success, and no matter what type of career is needed to achieve that success, communicating effectively is a major part of any career.

Because being an effective communicator is nonoptional, you're going to need to get past your fear. Section 12.2.1 discusses a path that worked for me and has worked for others as well.

12.2.1 Analyze the causes of your fear

We have a huge advantage as technology professionals because we're used to analyzing and troubleshooting problems. We do it with code, with networks, with operating systems, with data structures, and even with business processes. Most of us analyze and troubleshoot all the time without being conscious of it.

I was once discussing troubleshooting with a colleague. "I'm terrible at it," he said. "I always get stuck with where to start." It was a hallway conversation, nothing serious, and we were both headed to the same meeting. In that meeting, we were told that we wouldn't be getting the head count we'd requested for our team, which meant that we wouldn't be able to implement a new set of product features that we'd been looking forward to (and that solved some specific customer problems).

"Wait a minute," my colleague said. "We're already spending more than that amount of money on the contractors that make one of the software components we use. We've been saying for months that it would be cheaper to ditch that component and use an off-the-shelf one that we pay a small licensing fee for. Can't we do that now—use the new head count to integrate the new component and then put them on the new feature we know we need to build?"

"Yeah," I whispered sarcastically to him, "you're terrible at troubleshooting." He also thought he was a poor communicator, but from my perspective, both his troubleshooting and his communication were spot-on.

The point is that almost all tech people are really good at troubleshooting, even if we don't always recognize it. So let's troubleshoot the causes of your fear of communicating. Here are some possible causes that spring to mind, although you should make your own list (and don't limit yourself to these):

- I don't know enough to talk about this.
- I misspell things all the time.
- I stutter.
- I use punctuation incorrectly.

The key is to identify the specific root cause or causes of your fear. Don't say, "Talking in front of people makes me nervous," because that's too vague and does not get to the root cause. Why does it make you nervous? Don't say, "I don't like writing." Instead, ask yourself why you don't like it. Maybe you have a fear of writing because you took technical writing in college, and the teacher was harsh and unfeeling in his **critiques**. That's closer to a root cause: you felt that you were being made fun of, you didn't enjoy it, and you don't want to repeat the experience.

12.2.2 ***Address the causes of your fear***

When you start to identify some root causes honestly and frankly, you can do something about them.

Are you afraid of misspelling things? Fine: work on improving your spelling. There are online spelling-improvement courses and websites designed for adults.

Do you have a stutter that embarrasses you? Look into adult-oriented speech therapy classes.

Are you worried that you don't know what you're talking about? Remember the difference between ***confidence*** and ***arrogance***: ***confidence*** is knowing what you know, and ***arrogance*** is pretending to know something you don't know. Nobody wants to appear to be arrogant, but a lot of times, we downplay what we do know and consequently don't develop confidence. Work on developing your confidence in what you know. Don't be afraid that you may be wrong about something. It's fine to be wrong (or at least should be). If you work for an organization that punishes people for being wrong, you should seriously ask yourself why you work there.

DON'T FRONT-LOAD YOUR FEAR ***Don't derail your story by starting with a depressing remark about it.*** Whether you're writing or speaking, don't open with something like this: "Look, I probably am not an expert, and my concerns might not even be justified, so I apologize in advance if this isn't appropriate." ***Never apologize in advance;*** you undermine yourself as well as the communication you're trying to make.

Conquering your fear of communicating is a lot like troubleshooting. First identify the precise problem, and then address the problem. Your first attempt at addressing the problem may not solve the problem, but that's no different from troubleshooting code or network routing. Sometimes, you need to try different things before you arrive at a solution that sticks.

When it comes to debugging code or fixing a server, you're not allowed to just give up because you don't want to address the problem. It's your job to debug the code or fix the server, and you keep at it until you succeed. You hop on the internet to see what other people have done in similar situations. You take a best guess at the problem, try to fix it, and iterate from there. Solving for a fear of communicating follows the same process.

The only difference between troubleshooting and fixing a fear of communication is that your boss won't make you fix your communication. You have to choose to address communication fears on your own. It's up to you to develop the motivation and the sheer will to dig in and do it. The task isn't always comfortable, because it's not fun to look at your own weaknesses. But being an excellent communicator is part of your job and your career even if that work isn't written in the job description.

Recovering from failure

One way to increase your confidence and reduce fear is to have a plan for what to do when you fail. Having a plan means understanding and mitigating the downsides that failure represents.

For most businesses, the biggest downside of any kind of failure is repeating it. When something goes wrong, the big worry is that it'll happen again, which nobody wants.

My "recover from failure" plan is to make it clear that I learned from my failure and am unlikely to fail the same way a second time.

If I'm in a meeting, and I say something that someone else points out is wrong, I immediately acknowledge their correction, thank them for setting me straight, and then correct myself. I show that I'm open to failure, that I'm open to being corrected, and that it's safe for other people to correct me. I show that I'm capable of learning and then putting what I've learned to immediate use.

Having that plan makes me feel a lot safer about being wrong, because I know that everyone around me will see that I'm actively working to mitigate future failures.

12.2.3 Conquer fear in written communications

I find that it's easier to address weaknesses in written communications as a first step because writing isn't done in the moment; you have time to write, to edit, to reconsider, and to edit again. Here are some ways to improve your written communications:

- **Do not use autocorrect.** Instead, use your word processor's spelling and grammar check tools to call attention to your errors. When the software highlights a word or sentence, take the time to understand why it identified that text as a mistake. Research grammatical terms and proper spelling, if need be. **Don't let the software fix things for you; learn the fix yourself.**
- **Use a tool like Grammarly** (which requires a subscription) that can dig deeper into grammar problems than most word processors. Consider using one for a while to see whether it explains the problems and makes the fixes more understandable.
- **Take much of what you learned in college writing classes, and set it to one side.** Too many of those classes focus on formal types of communication, often requiring you to write in an **awkward**, stilted-sounding style. You may have learned this style of writing because that's how technical documents are written, but this is

why so many people hate reading technical documents. Instead, write like you talk. **While you write, imagine that you are having a conversation with the person or group who will receive your written communication.** Then, after you write a piece, read it out loud to yourself, and see whether it sounds like you.

It's okay to be in your communication

There's a myth—perpetrated by outdated writing classes in some cases and by all the other folks who've been trained in this myth—that you shouldn't exist in your writing. That is, it's not okay to use the word *I*. To go along with that, people are sometimes taught to avoid words like *you*, *we*, *me*, *us*, and other words that refer to specific people.

I prefer to write in a casual style that reads more or less like I speak. I use contractions, I refer to myself as *I*, and I refer to the reader as *you*. If the reader and I are going to do something together, such as follow along with a demonstration, I'll refer to us as *we*. This writing is easy to read; it's more natural, the sentence constructs are less awkward, and the writing flows more smoothly. Today, even academic journals—long the bastion of overly formal, passive-voice writing—are more accepting of natural-reading text.

I'll take that last item a step further; look at how I write right here in this book. I use contractions all the time (although sometimes copy editors who are following a specific style guide will remove them). I refer to myself as *I*, and you, the reader, as *you*. It's like we're sitting down together and having a chat. On the few occasions when I've turned one of my books into an audiobook and done the narrating myself, the work was easy, because my books already more or less read like a script for me. The best compliment I ever received was when a reader came to me after I'd finished a conference presentation and said, "Listening to you is exactly like reading your books, and now when I read your stuff, I'm going to hear your voice in my head."

If you're comfortable using short sentences, fine. Do that. Most people tend to speak in relatively short sentences, and short sentences can be easier for others to comprehend. You don't need to have a special style that you write in and a different one that you speak in.

Learn what passive voice is, and work hard to avoid it in your writing. I'm going to spend some time on that topic later in this chapter, as it's worth special attention.

When you feel comfortable with what I call the mechanical bits of writing—spelling, grammar, and writing something that reads naturally—you can get into the structure of your writing.

12.3 Applying structure to your storytelling

Structure brings us back to storytelling.

These kids threw the witch in the oven! Can you believe that? Well, I mean obviously she was going to eat them; she was the one who turned the oven on in the first place. And yes, I suppose they were eating her house, but who makes a house out of pastry and candy in the first place? She was obviously a predator trying to entrap children. She might as well have been driving a windowless white van.

Not such a great story, right? The writing is mechanically fine, but the structure is wrong. Who's the hero? What is their problem, and what is their journey? As you write (or rewrite), ask yourself, "Am I presenting things in a reasonable order that will make sense to my audience, or am I presenting things out of order? Am I providing sufficient context to make my story make sense to my audience, or am I going off on tangents?"

Creating a good story requires you to think about your audience, which many of us **don't do instinctively**. Consider this passage:

Our coding standards were developed in a different time and place. Originally, our goal was primarily to make our code more readable, so our standards focused on naming conventions for variables and functions. Over time, we developed basic standards for modularization to try to address the code bloat we were experiencing in some critical sections of the code. But back then, we all used a single coding language: C#. Today, the organization has evolved into a polyglot, with a large code base of C#, a large code base of JavaScript, and a growing code base of Python within the DevOps engineering teams. The coding standards we originally developed no longer scale well across these different languages, and attempting to maintain them slows us down and creates animosity between our teams. I suggest that it is time to rethink the purpose of coding standards, what value they bring to us, and how we achieve that value in our current environment. And rather than the top-down approach we used in the past, I would argue that a more collaborative approach would give everyone, and every team, a stake in the discussion.

Is that passage well written? **Take a few minutes to critique it.**

(Did you catch *were developed* in the first sentence? That's passive voice; you can tell because it doesn't say who developed the standards. I'd want to rewrite that as something more clear and direct, taking out the passive voice, such as "We developed our coding standards in a different time and place." If you can't see what I'm talking about, don't worry; later in this chapter, I'll go into the passive voice and its chilling effect on your writing.)

What you may realize is that you can critique it only to a point. After all, you don't know who the audience is. You don't know what context the audience already has. Is the discussion of where the coding standards started relevant? Maybe, if the audience doesn't have that history. But if they do, it might be time-wasting. The point is that communications should be designed for the audience who will receive them. Have you ever sat in a meeting with someone who droned on and on about stuff you already knew? They didn't design their communication correctly.

Beyond that, the structure of my example passage could use a little improvement. Along with the focus on the classic story structure of the hero's journey, effective written communication often follows a problem-context-solution approach, as I'll demonstrate in this rearranged version:

[Present the problem first.] *The coding standards we originally developed no longer scale well across our different languages, and attempting to maintain those standards slows us down and creates animosity between our teams.*

[Provide context.] *Our coding standards were developed in a different time and place. Originally, our goal was primarily to make our code more readable, so our standards focused on naming conventions for variables and functions. Over time, we developed basic standards for modularization to try to address the code bloat we were experiencing in some critical sections of the code.*

[Connect the context to the problem.] *But back then, we all used a single coding language: C#. Today, the organization has evolved into a polyglot, with a large code base of C#, a large code base of JavaScript, and a growing code base of Python within the DevOps engineering teams.*

[Offer a solution.] *I suggest that it is time to rethink the purpose of coding standards, what value they bring to us, and how we achieve that value in our current environment. And rather than the top-down approach we used in the past, I would argue that a more collaborative approach would give everyone, and every team, a stake in the discussion.*

If you're a visual person, you can think of it like figure 12.1.

I like this version of the original memo a bit more because it follows a clearer problem-context-solution approach. I want to point out something especially effective: by writing "We developed our coding standards in a different time and place," the author gives a face-saving out to anyone in the room who may have been a defender of the original coding standards. "Hey, it's not your fault," they're saying. "It was the right thing for its time, but we're in a different time now." In doing so, the author avoided a needless confrontation but didn't waste time beating around the bush.

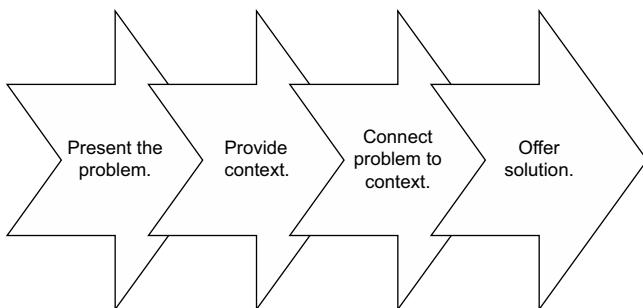


Figure 12.1 A model for effective communications

Problem-context-solution in action

I'll offer an example of this process in action, from a situation that occurred as I was writing this book.

In my company, we have a formal, structured process for creating new product designs. The process works well in most of the domains in which we work, but one particular domain is a little different. Our normal processes mainly involve speaking with customers to understand their problems and crafting products to solve those problems. But in this one special domain, a bit of science is involved, so the company employs people who are experts in those sciences.

The problem is that our usual design process was diminishing the voices of the experts we'd employed. Usually, we try to avoid the groupthink and bias that internal stakeholders can bring to a design, because we want to ensure that we're listening solely to our customers. But in this domain, our customers aren't aware of the science that's affecting how they work, and our experts have a bigger role to play in the design process.

Someone on the team decided to address this point by writing a short missive and took the problem-context-solution approach. This is what they came up with:

Present the problem. Our experts aren't being given a loud-enough voice in the process, even though their knowledge is essential in producing an effective product. We've seen this result in less-effective products in specific situations, and we should work to correct it.

Context. Although most of the problems with design solutions for require this special knowledge, in this one domain, we find that customers lack critical knowledge about the problem. Our experts, by leaning on a century or so of science, have that critical knowledge.

Connection. For this one domain, we need to modify our standard process so that we can amplify the voice of our experts. They can help us interpret what we're hearing from our customers and take advantage of the immense amount of proven science on this topic.

(continued)

Solution. I recommend that we begin including our experts not in the initial “talking to people” phase, but in the later “synthesizing what we’ve heard” phase so that they can help us better interpret what we’ve heard in the context of their scientific expertise.

This recommendation was ultimately adopted, and it was almost an “Ooooohhhh, I get it” moment for some of our team. By presenting the problem, providing a shareable context, connecting that context back to the problem, and then suggesting a solution, the writer created a space for understanding, a healthy shared discussion, and ultimately an improved workflow.

Focusing on both the mechanical and structural in your writing can be time-consuming, I know. But if you keep at it, you’ll gradually get better. Human brains are amazing things, and they’re wonderful at picking up on what we’re trying to do. If you practice this kind of writing enough, eventually your brain will start doing more of this structural planning proactively, without your having to think about it as much.

Practicing your writing is one of the easiest things in the world to do, thanks to the internet. Start a blog. Don’t worry if anyone else reads it. Heck, make it an entirely private blog, if you want. But dedicate yourself to writing in it at least weekly. Write about a problem you solved, a conversation you had, or any experience or idea that interests you. You’re not trying to teach something to anyone, so the topic isn’t important. What’s important is the act of writing—the process of reviewing and reordering what you’ve written and the focus on the mechanical aspects of writing. Just do it, and I promise you’ll get better at it.

12.4 Practice, practice, practice

There is no way to improve your communications skills except through constant practice. Our global culture doesn’t offer many informal opportunities to practice, though: we communicate more in informal writing like text messages, in which spelling isn’t a priority (let alone grammar). So here are some more suggestions for getting your practice in:

- **Blog.** As I’ve already suggested, blogging is a good way to practice. Write something, even if it’s only for you.
- **Commit to making your work and personal emails more structured.** Avoid one-word or one-line replies; write something more meaningful for the sake of practicing.
- **Even in instant messages, make an effort to level up your writing.** This includes business platforms such as Slack and Teams. Use complete sentences, tell a story, and strive to have writing that shines.

The time you spend improving these skills will repay itself a hundredfold when it comes to enhancing your career.

12.5 Common written defeaters

Writing can be error-prone, of course, and I don't mean just spelling and grammar errors. A lot of us defeat ourselves with our writing, making it an area in which almost nobody is perfect. But by focusing on two of the most common defeaters, you'll start pushing yourself closer to professional writing.

12.5.1 Avoid passive voice

If you already know the difference between active and passive voice, stick with me anyway for a second. There's more here than meets the eye.

You can read official definitions of active and passive voice at sites such as the Grammar Girl blog (<http://mng.bz/XYoG>), but here's how I think of them: in a sentence with active voice, it is clear who is doing what to whom, whereas with passive voice, an action gets done, but it's not clear who's doing the doing.

Consider *The computer is restarted* versus *Sandy restarted the computer*. The first example is passive, and the second is active: Sandy is doing the action (restarting).

College technical writing classes tend to emphasize passive voice, and many of my friends and colleagues tell me that this is where they learned to use passive voice so deliberately. The theory, they tell me, is that technical documents shouldn't refer to specific people; they should avoid using words like *you* and *I*. This approach implies that there are no people in technical documentation, only the things that happen. Can you imagine if technical writing professors wrote fairy tales?

The shoe is manufactured from glass measuring 9 on the Mohs scale. The shoe is worn only until midnight local time. If used during running, the shoe may detach from the wearer and be left behind. The shoe might then be found by someone else.

In real life, we like characters in our stories. Stories are almost always about people, so put people in the stories. It's fine to use words like *we*, *you*, and *I*. Write in active voice, and make it clear who is doing what.

AVOID THE "ROYAL WE" Use *we* only when you're referring to a group that includes yourself ("We need to think of a new solution to this problem"). Don't use *we* in an attempt to avoid the word *you* ("We will learn how to restart a computer").

One reason why I emphasize passive and active voice is that in running communications workshops, I've found that recognizing and understanding them can act like a switch to make someone's writing more engaging and natural. That is, when you stop using passive voice, something in your brain flips around, and you start writing wonderful, naturally flowing text that reads much more like a person speaks. Passive voice keeps people from writing like they talk, and as soon as they start using active voice,

feeling permitted to use words like *I* and *you*, they suddenly get it and start writing well. Give it a shot for yourself.

12.5.2 Prune that flowery garden

We all know that businesses love almost nothing more than trying to dress up language.

In trying to create synergy between our disparate bases, we have identified a number of blocking issues from our customers. The ask we have received is high level, but once we double-click into it we discover a variety of opportunities.

I honestly can't tell whether that is good news or bad news. This kind of business-speak not only hides the meaning, but also triggers my inner word **nerd**. Let's look at some of this jargon.

- **Issues** have no objectively correct or incorrect answer, only opinions that are open for discussion, such as political issues. Also, magazines have issues. Yet the use of the word *blocking* suggests that we're dealing with a problem. If it's a problem, call it a problem.
- **Ask** is a verb. It is not a noun. You cannot have an ask. You can have a request.
- **Double-click** is a trendy way of saying "look deeper into something." I suppose it was inevitable that *drill down* would be replaced by *double-click*, but it's already feeling **archaic**. Maybe this should be *double-tap*? I'm being only slightly sarcastic here.
- **Opportunities** are usually good things, but in this case I'm left wondering whether the word is a replacement for *problems*. It's impossible to tell, which means that this text is isn't doing the basic job of communication, which is to convey information accurately and unambiguously.

Just because your CEO likes to write flowery, ambiguous, misappropriated words doesn't mean that you have to do it too. I've always tried to advocate for clear, unambiguous writing. If we already have the word *request*, use it. If something is a problem, and you want to work on solving it, call it a problem. The title of this section, for example, is "Prune that flowery garden." Why didn't I use "Avoid ambiguous, pretentious language"?

12.6 Action items

For this chapter, I offer some exercises to help you improve your communications, with a particular focus on written communications:

- Start by going through some of your sent emails and direct messages. Given what you've read in this chapter, how would you rewrite any of those communications?
- The next time you send a substantial written communication (say, 600 words or more, which is about two pages in Microsoft Word), follow up with the recipient

and ask them what they thought of it. Was it concise? Did it communicate everything they needed to see? Was it clear and well ordered?

- The next time you need to make a written recommendation, review what you've written before you send it. Is it concise? Is your recommendation justified, with clear, unambiguous, well-presented data? Do you acknowledge other options and provide data-based reasons to not consider them?

13

Conquer verbal communications

In chapter 12, you built a foundation for effective communication in writing. Now it's time to use those skills to conquer verbal communications. I've always found writing to be a bit easier because I can usually take my time with it. It's not of-the-moment, right in someone's face, as verbal communications are. But many of the lessons of written communications are fully applicable to verbal ones, making writing a great starting point. Let's step up to speaking.

13.1 Stepping up to verbal communications

If you've really started working on your writing, your verbal communications will automatically become easier and less stressful. If you write like you speak, you'll be teaching your brain to organize your thoughts for you, and it'll do so in a way that makes it easier to speak as well.

If you have something formal to present verbally—maybe a team status meeting or some other fairly routine communication—start by writing yourself a script ahead of time. Read that script aloud to yourself ahead of time. Do multiple read-throughs, editing the script until the read-through feels effortless and natural.

At first, read from the script when you do your presentation. That might seem weird, depending on the group you're speaking to, so acknowledge it: "I've been working on my communications skills, and today I'll be reading from a prepared set of notes." If you've done enough read-throughs ahead of time, you'll be able to look up from the page now and again and make eye contact with people.

LOOK UP AS YOU FINISH A PARAGRAPH When I'm reading from a script in front of people, I'll look up and make eye contact for the last sentence in

each paragraph. My read-throughs give me the confidence to scan the sentence, look up, and recite it. Because it's the end of the paragraph, there's a white space after it, which makes it easy for my eyes to return to the same spot on the page afterward and not lose my place..

When you start gaining some comfort with reading from your scripts, you can start using notes instead. Still write the script, but don't take it into the meeting. Instead, take some notecards to remind you of what was in the script. Continue doing your advance read-throughs, and rely on the notes to remind you what to say. This process may be terrifying at first; push through. Over time, it will become more comfortable and natural. Eventually, you'll get to a point where you don't need the script at all: you can assemble the information as you speak. That may take years (it did for me), but it's worth the time.

NEED TO MAKE PROGRESS FASTER? If better verbal communications are an immediate for your career, consider joining an organization that focuses on public speaking skills, such as Toastmasters.

Finally, try to make eye contact whenever you're speaking. I almost always pick two or three friendly-looking faces that are evenly spaced around the room and rotate eye contact between them. They're usually people I know, people who are smiling, or people who in some other way don't look like they might eat me for lunch. It gives the illusion that you're looking around the room, even if you technically aren't.

13.2 **Conquering your fear of speaking**

Most tech people I meet don't relish the idea of public speaking. For some, presenting to an audience of more than one is terrifying; others are fine with small groups but hate being in the spotlight at the front of a conference room or on a stage. In talking with many people about this over the years, I've found that the fears boil down to a few root causes that actually aren't that hard to conquer.

13.2.1 **Fear of not having all the answers**

One of every instructor's biggest fears is a student asking a question that the instructor can't answer. It's a common fear in any kind of speaking, and it ties back to Imposter Syndrome, which I discuss a lot in this book. "If I keep quiet," our instincts tell us, "nobody will know I'm really the dumbest one in the room." That's something we all have to get past:

- **Nobody knows everything.** It's fine to speak about something that you aren't an omniscient expert on; nobody else is either. Review what you plan to speak about ahead of time, remind yourself to be confident in what you know, and forge ahead.
- **Be prepared to take notes.** If someone asks something you don't know, immediately jot it down, offer to follow up later with an answer, and move on. We all know that we don't all know everything, right? Admitting that you don't know, but

making a note and offering to follow up later, is the best anyone can ask of you. Make sure you do follow up, of course!

- **Try to anticipate questions.** This is an area where a lot of technologists I know struggle. I was asked to present to my company's executives (the so-called C suite, consisting of the CEO and all his direct reports) on a project that we'd been exploring. They were looking for a detailed recommendation, and I moved through the material I'd prepared fairly confidently. Then the CFO asked, "Are we sure we're going to be able to earn additional revenue on this if it's a product?" I was **dumbstruck**. My executive had asked my team to research this and present a recommendation; it had never occurred to me that I'd need to know whether or not the product would . . . you know . . . be worth it or not. I admitted to the CFO that I didn't have that information and that I'd follow up later. After the meeting, I worked with some other teams to find an answer.

Perhaps most important, I learned a little bit more about how the business operated. In future presentations like that, I knew to anticipate those questions and have answers ready. That experience—like almost any experience where you realize you don't know something and then work to find out—helped make me a better businessperson as a whole.

13.2.2 **Fear of being judged**

Obviously, none of us wants to look stupid or silly in front of our peers, our bosses, our colleagues, or anyone, right? Except for one or two unfortunate examples that I've come to realize were real outliers, I've come to understand that most people are rooting for other people. Your peers, your boss, and your colleagues generally want to see you succeed. Maybe they're not great at showing it; sometimes, they're just as afraid as you are of letting their body language show their support, for fear that they'll be judged, but their support is there.

Don't worry about being judged. Instead, build up your skills and confidence. I've probably talked with a thousand other technologists and asked them how they **overcame this part of their fear of speaking, and their many answers boiled down to one:** practice. **The more you speak in front of other people, the more comfortable you'll become, and the more you will come to understand that your audience is genuinely cheering you on, even if they're doing so silently and stoically, behind a poker face of sorts.**

Organizers like Toastmasters exist in large part to provide an opportunity to practice, but you don't need a formal organization to get your hours in. Hard as it may be, volunteer to do brief presentations at work. Force yourself to do it, despite your fear. Eventually, I promise, it'll become easier and easier. As your fear dissipates, you'll be able to focus more of your attention on refining your speaking skills until you're comfortable, calm, collected, and effective as a verbal communicator.

13.3 Common verbal defeaters

Few people are perfect orators. Most of us have brains that seem to operate at different speeds from our mouths, and most of us have anxieties and fears that hold us back. The good news is that you don't need to be a perfect orator. You only need to be good enough. If you focus on three basic, straightforward practices, you'll get to more than good enough quicker than you might think.

SQUASH THOSE VERBAL, UM, FILLERS

The next time you're speaking to someone, ask them if you can record the conversation for your own later analysis. Count the number of times you say "like," "um," "uh," "ya know," or any of those other verbal fillers. Then go look up a really good public speaker. Specifically, go look up a speech by former US President Barack Obama. It doesn't matter what you think of his politics, and you don't even need to listen to the words he says. Pay attention to his speech pattern. This is a little hard to convey in writing, but Obama . . . paused . . . a lot. I've wondered whether he had a problem with "um" or "uh" earlier in his life.

The little verbal fillers happen when our brains need to stop for a half-second to catch up. For some reason, we believe that we have to fill every single moment when we're speaking, so we insert a filler: it's like when, uh, we just need a break.

Instead of using the verbal filler, train yourself to pause. It does not sound weird, trust me. In fact, it gives your audiences' brains a chance to catch up as well. The pause will sound unnatural to you at first, but go watch any TED talk video and tell me how many times the presenter uses a verbal filler like "uh." It'll be approximately zero in most cases, but you'll hear plenty of verbal pauses where an "uh" could have gone.

When you start this process, you're going to trip yourself up a lot. You're going to become hyperaware of the verbal fillers. I once caught myself saying something like "So we're going to look at the, um, oh sorry, the um, the database console, sorry." That wasn't a fun day. But you do get past it, I swear. And when you do, your speech will automatically sound less fragmented, less chaotic, smoother, more planned, and more professional.

USE YOUR VOICE AS AN INSTRUMENT

Humans have been living together as a species for a long time. As a result, we've developed a great many nonverbal means of communicating: facial expressions, general body language, hand gestures, and so on. Those complement our verbal communications, lending nuance and emotion to what we're saying. But we've also developed a subverbal language, meaning that it's not just what we say that communicates, it's also how we say it. Think about the different ways you could say "Oh, that's nice" to someone, and you'll start to understand the concept.

Listen to the YouTube video at <https://bit.ly/CompareSiri>, which compares the original voice of Apple's Siri assistance with a newer rendition from several years later. The original voice is flat and robotic, placing slight emphasis in strange spots. The new voice is better, but it's still light years from a human voice. It gets the emphasis

wrong in places, and it still raises the pitch of its voice in odd spots. **You don't want to be Siri when you speak.**

I have a friend who hates the sound of his voice. He dislikes it so much that he refuses to let anyone record him saying anything. I've been in the room with him when he was on a conference call for work, and I kind of see where he's coming from. **His work voice is extremely flat and emotionless.** It's like someone once told him that work speaking should be emotionless and robotic, and he embraced that. He still sounds better than Siri, but his voice isn't as engaging, compelling, or human as it could be.

The sad thing is that it's all an act. He doesn't talk like that around his friends, and I suspect he doesn't even talk like that around the office. In those situations, his voice rises and falls naturally. The pitch of his voice rises when he asks a question and falls slightly when he's making a firm statement. In the culture he and I share, those are the normal and expected speech patterns for an adult. But in a formal speaking engagement, such as a meeting, he goes all flat. He speaks in something close to a monotone, the pitch of his voice varying only slightly. **It's eerie.**

It reminds me a little bit of being asked—or forced, depending on your perspective—to read aloud when I was in grade school. Some kids loved it, adding slightly different voices to different characters' speech. Other kids clearly didn't enjoy it, reading in a robotic monotone that made it hard to assemble what they were saying into sentences.

Your voice is a powerful, beautiful instrument. The brains of your fellow humans are preprogrammed to react to that instrument, to feel certain emotions when you speak in a given way and to respond to subverbal cues like the pitch of your voice.

I've found that one way to get better at using my voice is to first be comfortable with the content I'm planning to present. That's why I do read-throughs in advance. Sometimes, for a really important presentation, I'll mark up a script:

Good morning. //

Today, I'm going to address a critical problem that we've been dealing with for a couple of months now, /// and focus on three potential paths forward_. //

The problem is one of data security, // which is something our executive team / has identified as a top priority for this quarter_. //

As many of you know, // our current data management tools are simply overwhelmed / with the amount of data we have to manage. // Customer data, / internal analysis data, / and product data are all growing at exponential rates_.

I use /// slashes to tell myself where to pause. The more slashes, the longer the pause. For three slashes, that's a long enough pause to quickly scan the room with my

eyes. Underlined words are ones I want to lightly emphasize; I use *italics* for a stronger emphasis. The _ underscore reminds me to let my voice pitch stay even or drop slightly; I usually only want my voice pitch to rise at the end of a sentence if that sentence poses a question.

My *italics* are often ways of sharply highlighting particular words or word pairs. If you were to imagine the above passage having an accompanying PowerPoint slide deck, then you'd be looking at a slide that said Problem: Data Security. I'd probably then slide in bullets for the three types of data I mention.

So briefly, here's my methodology, and you should consider constructing a markup methodology that makes sense to you.

Markup	Meaning
<i>Italics</i>	Heavier emphasis
<u>Underline</u>	Lighter emphasis
/ / / / /	Pauses—the more slashes, the longer the pause.

That passage takes me about 35 seconds to read through. Try timing yourself reading that aloud, placing all the emphasis and pauses in place that I've noted. If you're much faster than 35 seconds, you're probably rushing it. Slow down. Remember, your voice is an instrument, and people need time to hear and react to it. Practice putting emphasis in different places, and even record yourself. If you don't like how you sound . . . then fix it. Practice until you *do* like how you sound, because other people will probably like it a lot as well. Practice, practice, practice.

PRACTICE STAYING ON TRACK

I have a dear friend who seems to be biologically incapable of staying on track. This is a common example of our conversations:

FRIEND: So you know when I was out hiking? It was Sunday—or no, was it Saturday? We went to brunch Sunday, but I thought Saturday was too hot. Was it—

ME: Does it matter?

FRIEND: It might have been Friday. Anyway, I was out with Angela, and she brought her sister along. Mary? Marsha? She's the one from Iowa. I think it was Mary. Hey, [turns to other friend] was it Mary? Angela's sister, the—

ME: Does it matter? Let's call her Mary.

FRIEND: Anyway, we'd just got back from the hike, and Angela drove. She's got one of those new Kia SUVs, the Yukon—

ME: Yukon is GMC.

FRIEND: No, it was definitely a Kia, because—

ME: I'm going to the bar.

Our conversations are notable only when they *don't* have a dozen irrelevant tangents. In your professional communications, you want to try and avoid that kind of behavior.

I find it helps to really focus your mind, right at the outset, on the *outcome* of what you're about to say. What is the goal? In the previous example, it turns out my friend's goal was to point out that Angela's new SUV had a heads-up display that projected the current speed and other information onto the windshield right in front of the driver. That fact took a full 20 minutes to get to. It was fine; we were at a bar talking. But as a professional communication, my friend could have started with "The new Kia SUVs have a heads-up display" and basically ended there as well.

Remember: communicating is *storytelling*. You want to get to the moral of the story pretty much as quickly as possible, with as few side paths as possible along the way.

13.4 Finding the right amount of assertive

When it comes to communicating, whether in writing or speaking, I still struggle with how to be the right amount of assertive. I happen to be the kind of person who has strong opinions, and I don't mind sharing them in the right place and at the right time. If my boss asks a question about how our team should be doing something, I'm often the first one to offer options. So I don't have a problem being assertive. But sometimes, that can come off as *too* assertive, making other people feel like there's no room in the conversation for them. I certainly don't want that.

VERBAL ASSERTIVENESS IS THE MOST NOTICEABLE While it's certainly possible to be over-assertive in writing, I think verbal communications are where assertiveness becomes the most noticeable. Our tone of voice, our body language, and our simple passion for something can come across much more easily in verbal communications.

Other people, however, do have difficulty asserting themselves—oftentimes because they're simply more polite than I am. And some other people have even less of a problem being assertive than I do, often coming across as aggressive and pushy. So there's definitely a spectrum, as shown in figure 13.1.

I think most professional people generally need to aim for the middle of the spectrum: when you have an informed opinion on a subject, bring it to the table. Be open to discussing it. Listen to pros and cons, and be open to evolving your opinion based on the facts presented to you.

At one wrong end of the spectrum is the overly aggressive person who pushes only for their perspective, who is not open to facts or reasoned objections, and who

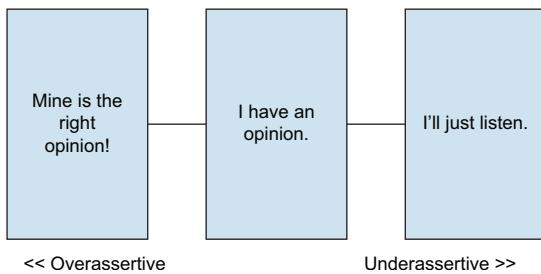


Figure 13.1 The spectrum of assertiveness

dominates the conversation with their opinion. Sometimes, that kind of over-assertive attitude is cultural: there are places where it's not only socially acceptable to behave that way, it's practically mandatory if you want to get anything done. To those people, I suggest they express, don't repress. Your opinion is valuable, I tell them, but it is not the only opinion. Your co-workers' opinions are also valuable, and those co-workers deserve a chance to be heard without having to shout you down. If you're seen as a repressive communicator, you'll damage your career and your professional network.

At the other wrong end of the spectrum is the under-assertive person who rarely offers their perspective. Their reticence often comes down to Imposter Syndrome: they're afraid to speak up, because everyone else in the room is so much smarter than them. Or sometimes, the under-assertive person was just raised to guard against being over-assertive, and they're being a little more cautious than they need to be. Or, perhaps the under-assertive person just isn't interested in a yelling match with an overassertive person. Regardless of where it comes from, I offer this advice: express yourself. The fact that you're in the room at all means you earned your place there. You not only have a right to contribute to the discussion, you have an obligation to do so, because it's part of what you're being paid to do as your job.

Whomever you are and whatever you do for a living, practice keeping yourself in the middle of the spectrum when communicating. Make room for other people's opinions, and support others for offering their perspectives. Put yourself into the discussion, and offer your colleagues some trust that they'll welcome your participation.

13.5 Persuasion and the art of listening

One of the main reasons humans communicate is to exchange information and ideas; another main reason we communicate is to persuade each other. Persuasion is not a bad thing: if you and I are working on a problem, we might each have different ideas about how to solve it. In a healthy working relationship, I might persuade you to see how my way would best solve the problem. You might try to persuade me that your way would be best. So long as we can both join in a fact-based, data-driven discussion, and so long as we are both open to being persuaded by facts and data, our debate is a healthy one that will likely lead to the best possible outcome given what we know.

But the trick about persuasion is that it requires the active participation of both parties. If you're not listening to me, I can't persuade you of something. If I'm suggesting

something that you know is physically impossible, and if I'm not listening to your objections, then I'll never persuade you. We might argue a lot, but we'll never really get anywhere.

Here's how to get someone to actively participate in being persuaded by you: listen to them. Let's suppose you're trying to convince your boss to purchase a new software tool that will help make everyone's job easier and faster. Your boss is objecting:

The money for this wasn't really in the budget. I could probably find it, but I can't just do this on the promise of increased productivity.

Your boss has just told you everything you need to know to persuade them. What the boss did is express their particular perspective, and their own motivations. Once you know their perspective, what they care about, you can start crafting an argument that speaks to their concerns.

The boss cares about the budget, but the price alone isn't the objection in this case. What the boss really cares about is productivity, and being able to prove that the expenditure increased productivity. It sounds like there may have been past instances where investments of this type didn't create a positive return, and so the boss is a little worried about that happening again.

Notably, the boss didn't express an interest in making everyone's job easier. That's often the case: everyone is getting paid the same, and "easier jobs" doesn't always translate to "better outcomes." Now you might be thinking, "Sure, but if our jobs are easier, we can do more work." Not untrue, but you need to express yourself using the terminology of the person you're trying to persuade. If the boss uses the word productivity, use that word. Understand what it means, and how it's measured.

Actually, boss, this tool should help us increase our number of successful builds per day by about 25%. I've spoken with friends at other companies who use this, and it's just wiped out all of their manual labor so they can focus on coding. We measure productivity partly on successful builds per day, right?

Persuade people by listening and by giving them answers in their terms, that they care about.

13.6 Action items

Here are some exercises to help you practice and improve your communications, with a particular focus on written communications:

- Create reasons to hold short verbal presentations—maybe just a 10-minute briefing in front of your own team, for example. Ask someone to record you

speaking (a smartphone recording is fine), and review that afterwards. Given what you've read in this chapter, what might you change about your presentation? Continue working through small presentations until you feel yourself becoming more comfortable and effective.

- Ask a colleague at work to score your next verbal presentation, whether it's just a small meeting with your team, or an important presentation in front of the bosses. Ask them where they felt you were compelling or not, where they think you stayed on-track or not, and how well you avoided using verbal fillers.
- The next time you speak with a group—even if it's just your team—ask them if you can record it, so that you can listen to yourself later. Critique yourself, and find one or two specific things to work on improving.
- Listen carefully to other speakers, both good and bad. Make notes immediately afterwards: What did you like about them? What didn't you like? Do you see any of those "don't-like" characteristics in your own speaking? What might you learn from those other speakers to help improve your own verbal communications skills?

Resolve conflicts

Anytime we're at work, we're with other people. That automatically creates opportunities for conflict: **we all have different opinions, different priorities, and different perspectives, and when those don't mesh, conflict can arise.** Conflict can actually be healthy at times, provided you know how to work your way through it with grace and professionalism.

Before we dive in, let's acknowledge some of the different types of conflict that can arise, since we'll need to take slightly different approaches to each:

- Businesses often create deliberate, intentional tension that's designed to help weigh competing interests and priorities within the business. When used properly, these create "healthy conflict" that helps businesses navigate tradeoffs and compromises.
- Businesses can also have unintentional conflicts around business decisions, which are often the result of **unforeseen circumstances**.
- As humans, we can create or be subject to personal conflicts. These may arise from business situations, such as two people disagreeing on a particular decision, but they're almost always unhealthy and almost always need to be resolved.
- Finally, we humans can also have interpersonal conflicts. These can happen for all kinds of reasons, which we'll explore in this chapter. Interpersonal conflicts are the ones that tend to affect us most personally, and we may not even want to raise them at work, but they're the conflicts that can create the most damage to a team, and to our professional brands.

14.1 **Conflict can be healthy and even deliberate**

We often think of conflict as a bad thing that has to be dealt with, but sometimes leaders encourage very deliberate, intentional conflict that can be healthy for the business.

For example, I once wanted to open a new position on my team for an experienced technologist. We'd been relying on contractors for months, and I could see that the contractor was costing us more than even a fully loaded salary would cost us. Having a permanent employee would save us money, bring more stability to the team, and—with the right hire—bring more passion and perspective than a contractor might. Our Finance team, on the other hand, flat-out refused to approve the new hire. Boom! Conflict!

Many businesses refer to that kind of situation as deliberate tension. Businesses have a lot of factors they have to consider in making decisions, and they often choose different departments, teams, or people to represent those sometimes-conflicting factors. Finance's job was to protect the company's purse, ensure the company maintained an appropriate profit margin—a key metric that the market used to determine the company's health—and monitor the company's cash flow. My team was responsible for creating customer-facing outcomes, such as products and services, in an efficient and effective way. In the instance I'm describing, those two perspectives were in conflict, and that was very much on purpose. The company's executives trusted that, between us, Finance and I would sort through the appropriate motivations, drivers, concerns, and outcomes to find the most well-balanced answer possible. There was no objectively correct answer in this situation, which meant Finance and I would need to work together to find the best balance we could, making the best set of tradeoffs we could agree to. We'll come back to this story throughout this chapter, to illustrate several key points about conflict, and how to handle it.

Conflicts can also be unintentional, such as when two co-workers disagree over a decision that needs to be made. The disagreement might be over something as seemingly simple as how close everyone's desks should be, or something serious, like choosing a programming language for an upcoming project.

Whatever the situation, the conflict resolution process actually works out to be really similar. In fact, when you find yourself in a personal conflict with a co-worker, one way to take the stress out of the situation is to treat it like one of those deliberate "business tension" conflicts like the one I was having with Finance.

14.2 **Seeking context**

I touched on this exercise in chapter 10, but I want to revisit it here with a different intent and outcome in mind. So after reading this paragraph, I'd like you to close your eyes. Sit back, relax, and for 5 minutes or so, just think. Don't even try to think about anything in particular; just see what floats through your mind. Go.

What came into your mind? Probably some work-related stuff, maybe around a project you're working on. Maybe something inspired by this book. Likely something to do with your personal life. Perhaps a snippet of song lyrics you heard earlier.

All of those things are your context. They're the things going on in your life right now, the concerns in the forefront of your mind. Your context also includes longer-term concerns: you might be worried about paying your bills this month, or thinking about the doctor's appointment you have coming up. Everything going on in your head, right now, is your context. Our context can play an enormous role in creating conflict, and in how we deal with that conflict.

Imagine for a moment that you walk into work one morning. You're in a decent mood: the commute went reasonably well, the podcast you were listening to was interesting, and you're looking forward to the project you're working on. Things are going well at home, too: there are no major financial or life problems facing you, the kids seem to be doing well in school, and your spouse is happy at work. You're especially looking forward to talking to Lisa, one of your co-workers. You and she have been working through a functional specification for a technology pilot, and today you were both planning on nailing down some of the final success criteria for the pilot.

As you walk into the office, you stop in the break room to grab a cup of water. Lisa's already there, waiting for the Keurig machine to pour out a cup of coffee.

"Hey, Lisa!" you say in a friendly voice. "I was thinking about the success criteria all night, and I think the user satisfaction score is really going to be the ultimate measure of success on this one. What do you—"

"That's stupid," **Lisa snaps, turning around to glare at you.** "Users have no idea if they're happy or not. We can discuss it later."

You realize that Lisa's jeans are torn, and she's covered in mud. Her glasses look like they might be broken, as they're hanging askew on her face.

In this instance, it's pretty easy to understand why Lisa might be in a bad mood: her commute obviously didn't go as well as yours. In other words, some of her context is publicly visible. You immediately mumble an apology and back off, wisely realizing that Lisa's not snapping at you, she's snapping at life. Today might not be the day to nail down the pilot's success criteria.

The problem is that ordinarily our contexts aren't on public display like that. What's going on in our world is all in our head, and to everybody else we look like we always do.

I once got into a fairly explosive argument with a co-worker about the programming language we were going to use for a new project. We didn't have a lot of internal software development going on, and most of what we did have was on a midrange computer; the project in front of us was a PC-based application, meaning we could go in whatever direction we wanted. He and I were both arguing for different languages, and it started getting pretty ugly. The language I wanted was stupid and for babies, he told me. Your language is hard to maintain and takes longer to code in, I argued.

Finally, exhausted by it all, he asked, "Look, what's your problem here? What's actually driving you to push for this language? Just tell me. If you can help me understand it, I'll back you up."

That made me pause. It took the wind completely out of my sails. I thought about it. "OK," I told him, "I've never actually done software development at this level

before. I know scripting language, and the language I'm pushing for resembles the scripting I've done. What you're suggesting looks like C++ to me. I just don't understand it, and if we go with it, I'm afraid I won't have a place on the team."

I want to pause the story for a moment and point out that I worked on a team that had made it safe to be vulnerable. It was okay to admit weakness. And this co-worker in particular had always made it safe to "look weaker," by never taking the opportunity to poke fun, make jokes, or otherwise create a less-safe environment. Safer environments always make it easier to resolve conflicts, which is why it's so important to invest in creating and maintaining a safe environment. But if you work in a company or on a team that does not make it safe to share your vulnerabilities, you need to be cautious about revealing what may be perceived as your weaknesses.

Okay, back to the story: I'd just shared part of my context. My co-worker now had more of an idea why I was pushing the way I was. "My concern," he said in a much calmer voice than we'd both been using, "is that there are way more professional programmers out there who use the language I'm pushing for. If we need to grow the team, it's going to be a lot harder if we go your direction. And frankly, I don't want to learn your language. It's not going to do anything to further my career. But you'd be doing your career a favor by going my way, and I'm completely willing to help you through it."

Now he shared some of his context, and also offered the beginning of a solution. We just needed to get inside each others' head for a minute.

Let's take this lesson about using context to help resolve conflict and go back to the story about my problem with the Finance department. "Okay," I asked my colleague in Finance. "Help me understand the parameters you guys have put up around adding new headcount." In other words, I asked for context.

"It's a lot of things," she said. "We're running really close to our target gross margin right now, and adding new people might put us below that target, since headcount is considered a cost. In this case, your hire might actually reduce expenses, which is good, but we're also looking at long-term cash flow. Frankly, we can let your contractor go with a month's notice. If we hire someone though, we morally can't just fire them if cash flow goes down. We're stuck with them, and the investors are really pushing us to maintain our margin and cash flow numbers. If we don't, it's going to hurt the next round of investment, and that could sink us."

"We're really that close to the edge?" I asked.

"We are," she said. "And we'll be fine, but we have to look really hard at adding to payroll. Why is this coming up? Is the contractor not working out?" Now she was asking for context.

"They mostly are," I said. "But they're several time zones off, and it's making collaboration harder than we anticipated. They're not able to be on all of our stand-ups, and they're working on projects for a couple of other customers at the same time. They're getting the work done, but it's just very mechanical. They're not invested in the project, and so we're not getting the benefit of someone's mind being all-in on it."

I just worry that we’re not creating an outcome that’s going to serve the customer like we want, which means we’re going to spend more cycles going back and fixing, rather than moving forward.”

She nodded. “I get it. And how long do you anticipate this project to take?”

“At least two years,” I said. “We’ve already got ten releases mapped out, and we’re working in eight-week sprints.”

“Okay,” she said. “It feels like we’re balanced on the fence between the competing priorities. Let me take it to my boss. I feel like the hire is the right thing to do, if we can make sure it’ll fit the financial model. Let’s go talk to HR and confirm what the fully-loaded salary would be.”

By seeking context from each other, we learned more about the situation, priorities, and pressures that each was dealing with. It was easier in that case than my more personal-conflict story, because we didn’t especially need a “safe space” to share our context. But the process and intention was the same.

14.3 **Returning to first principles**

I find that one of the best ways to start resolving a personal conflict is to try and take the personal out of it. Start the process by going back to first principles, meaning: Why are we doing this thing in the first place? Whatever we’re arguing about, whatever we both believe should be done or not done, we need to go back to this: What’s the point of it all?

I helped found a nonprofit called The DevOps Collective, and one of its activities was to put on the PowerShell + DevOps Global Summit conference. Several years in, I and my co-founders were in the process of stepping away from the organization, handing it over to a new generation of volunteers. It was important to us that the organization be able to outlive our involvement, and so we felt it was time to start bringing in new people.

We—the founders—almost immediately got into a conflict with the new folks over the topic of exhibitors at the conference. We old-timers had always shunned a formal expo hall like those at so many conferences; we felt that our event was about community, about learning, and about networking—we didn’t want it to turn into a commercial event like the giant vendor conferences.

The argument went on for quite a while, with both sides tossing “facts” back and forth. “Vendors are every bit as part of the community as individuals,” one person argued. “As soon as we accept vendors, we’re beholden to them, and we’ll start changing the schedule to force people to wander through the expo hall,” someone else argued.

“Stop,” someone finally said. “Why are we hosting this conference?”

We all thought about it for a minute, and came up with several reasons. “So we can educate. So we can all have a place to come together in person. Because it strengthens our community.”

The person who’d asked us to stop nodded and said, “Those are good things, but they’re not why we *started this in the first place*. Don,” they said, turning to me, “why did we start this conference?”

“To make money,” I said, after a bit of thinking. “The goal of the nonprofit is to help make tech skills accessible to young people living in situations where they can’t access tech skills. We need money to do that. We need money to pay for the PowerShell.org website. We want money to help bootstrap regional one-day events, for the people who can’t make it to the annual conference. The conference is valuable, and it needs to be valuable for people to pay for it, but the *reason* we did it is because it’s about the only kind of activity we could think of that we could make a profit on. We use the profit to pay for the other things. The education and networking are why people pay to come, but registration is touch and go every year. Sponsorship funds would help give us a bit of padding.”

Everyone stared for a minute.

“I get it,” I continued. “I don’t want the expo hall either. But the new people are right: it’s a way to help secure the financial future of the organization, and that was the point. Maybe we should accept that and start thinking of ways to include vendors in a healthy way.”

And we did. Well, not at first—to be clear, we made several missteps along the way. But we learned each time, and got better each time. Going back to first principles—why we started doing the thing in the first place—let us all align our *contexts* to that, let us move past the conflict, and helped us move forward in a healthy way.

14.4 **Relying on data**

Another way to remove the personal from personal conflicts is to step away from subjective opinions and instead rely on objective data. When I was working as an independent contractor, I had an opportunity to work with a company that was developing a mobile point-of-sale system. I showed up at a design meeting one day to find a knock-down, drag-out argument already in progress, over the design of one of the system’s main screens. “Whoa,” I said when there was a break in the yelling. “Can someone fill me in?”

“David,” Erin said (these are not their real names), “seems to believe that *this* is the best layout for the item selection screen.” Erin threw a mocked-up screen design on the room’s big screen. “I keep telling him that this clustered layout will *not* work. *This*,” she continued, switching to another mockup, “makes far more sense.”

“Okay,” I said, settling into a chair. “David, tell me why yours works better.”

“It’s more elegant,” he said simply. “We’ve already been told that each category has a different color button, and this clusters everything by color. It’s easier for your eye to go to whichever color you need.”

“If you memorize which color means what,” Erin snorted. “My layout uses a grid. Each column is a color, so your eye can still be drawn. But it moves the most common items to the middle, which is where your finger already is when you’re coming from the previous screen. So it’s more efficient in terms of movement. Daniel’s would have you—”

“Hang on,” I said. “What is it you’re both stating, here? Belief, opinion, theory?”

“Fact,” Erin said confidently.

“Awesome,” I said. “Show me the data.” She blinked. “If you don’t have data as proof, then it’s not a fact,” I reminded her. “You’re stating a theory. But the good news is that we can prove this out.”

Both of them had stopped breathing so heavily and were looking at me.

“Let’s spend a day on a functional mockup. You can each have a software engineer, and all we need it to do is register which buttons were pressed and how long it took. The rest of the team and I will take the day to come up with several orders, based on the customer ordering patterns we already have. Whichever design lets an operator punch in those orders fastest, wins.”

Everyone was happy with that. We’d managed to do a couple of things. One, without explicitly saying so, we’d come back to first principles: our job was to make order-entry as fast as possible, not to produce something that was “elegant” or “logical.” We’d also found a way to set our opinions aside, and focus on objective *data*. We’d stated a couple of theories and set out to prove or disprove them, using data. Data doesn’t have opinions. In the end, we found that neither Daniel nor Erin’s screen layouts were especially efficient. Erin’s was faster, but it was still slower than the older system we were meant to be replacing. So we continued iterating, and relied more on data to move forward.

14.5 **Using decision-making frameworks**

Another way to help resolve conflicts, particularly around business decisions (as opposed to personal, non-business-related conflicts), is to fall back on a decision-making framework. In chapter 15, I outline one decision-making framework called RAPID.

The goal of RAPID, and frameworks like it, is to make business decision-making faster, but it also helps mitigate conflict by clearly identifying everyone’s roles. You and I may disagree with what should be done, but if the two of us are an Input in the framework, then the ultimate decision isn’t ours anyway. As Inputs, our job is to provide information and potentially make a case for having the decision go a particular way. Ideally, we’d do so in as data-driven a way as possible, remembering the first principles of why we were all doing whatever it was in the first place. The person who owns the Decision would take that and make the ultimate call on what to do. You and I might be in conflict, but *they’d* resolve that with their decision.

I’ve found that using RAPID helps prevent aggressive conflict almost entirely. When I’m an Input, the other people acting as Inputs know that nothing’s personal. Nobody’s going to “win.” We’re all playing a role, in a specific situation, and the “owner of the D” tries to get as much data-based information as possible from us. We’re not *arguing*, we’re *assisting*. It’s one reason I’m such a big fan of having these decision-making frameworks, even if it’s just at the team and department levels.

14.6 A win doesn't matter as much as the outcome

In business, it's important to remember that everyone's there to achieve the same basic goal: serve the customer. In the details of the day-to-day activities, it can be easy to sometimes let our personal passions and perspectives carry us away. Once we get into a conflict, there's a very natural human urge to *win*: losing makes us look like failures, and makes us look weak. The weak get eaten! Only the strong survive! But we have to face that tendency and set it aside. *Winning isn't as important as achieving the best outcome for the customer.*

I sometimes will remind myself of that fact out loud, so that my colleagues can hear it, and remind themselves as well. "You know what, we're on opposite sides here, but none of us really care about winning. I know we all care about the best outcome. So: back to first principles. Why are we doing this thing? What *is* the outcome? What data do we have to guide us? What contexts aren't we considering?"

That, I've found, is the beginning of a great path toward professional conflict resolution.

14.7 Action Items

For this chapter, I obviously can't put you into a conflict. Even offering a story for you to "solve" misses the whole point of conflict resolution, because you'd have no way to get into the context of the characters. So instead, I'd like you to spend every day over the next week simply examining your *own* context. What's happening in your mind? What's influencing how you feel, and how you react to the potential conflicts that pop up at work? Are you reminding yourself to seek out first principles, and to look for objective data points—even if they don't support the position you started with?

15

Be a data-driven, critical thinker

Good businesses attempt, as much as possible, not to run on gut instinct or opinion. Instead, they try to be *data-driven* in as many aspects of the business as possible. Being data-driven can be difficult: as humans, we are strongly guided by our experiences, and the opinions those experiences help us to form. So even in ordinary, day-to-day, team-level thinking and decision-making, it's important for us to step back and let data be our guide.

15.1 **In business, never “believe”**

I can't speak for other languages, but in American English we use the word *believe* a lot. I actually find it a little problematic, in part because I like to split hairs over words, but in part because the word is not specific enough.

For me, the word *believe* refers to something that I accept as a fact, even though I have no data, evidence, or other proof of it being true. Religion is something that involves *belief*, for example. There is nothing wrong with believing in something. However, even I tend to use the word more *colloquially* at times: “I believe it's going to rain.” Technically, if I'm looking at a giant storm cloud and the barometer is falling, I don't need to *believe* in the rain. I have some evidence that it's coming.

That's why, in my working life, I try very hard not to use the word *believe*. After all, the business isn't (or shouldn't be) interested in what I accept as fact without proof, right? The business should be interested in what I'm thinking based on the facts and evidence around me. I don't want anyone at work to accidentally assume that I'm proposing to make decisions not based on facts, and the word *believe* suggests that I might be. So to avoid any potential confusion, I try to use other words.

If I have a set of evidence that seems to be pointing in a particular direction, I will express a *theory*: “I have a theory that we need to change our product in this way.” Theories can be debated, and they can be tested. I can work to assemble additional evidence to prove or disprove my theory. Using the word *theory* indicates that I’m willing to engage in the prove-or-disprove process, and that I’m not anchored to my statement as an indisputable fact.

I also feel free to state opinions: “In my opinion, we should refactor the code into the following modules, and here’s why.” My opinions are formed in large part from my experiences, which count as data points. I can share those experiences with others, and we can debate their applicability to the current situation. We may not be operating from hard facts, but we’re attempting to learn from history, which is always a great idea in business. Using the word *opinion* indicates that I may not be working entirely from hard facts, but that I’m integrating my own experience, as well. I’m fine with other people having differing opinions, and we can discuss them.

I will also state facts: “Our user satisfaction score dropped ten points last quarter.” Factual statements are backed up by data, and they form the strongest basis for forming theories and opinions. My team and I might debate the veracity of that data—essentially expressing a theory about its validity—and we can then work to prove or disprove it. Using the word *fact* indicates that I’ve accepted a data point as objectively true. That lines up everyone else to either also accept the fact, or to challenge its veracity, either of which lets the team move the discussion forward.

These are, for the most part, the only statements I’m comfortable making at work, especially when decision-making is involved: theories, opinions, and facts. I try to steer clear of beliefs, because others may have different beliefs. Because beliefs aren’t necessarily rooted in objective, shared data, it’s difficult for people to debate beliefs and apply them to the world of business.

15.2 **Be a data-driven, critical thinker**

Critical thinking is an exercise in which you deliberately try to remove your own filters, biases, and beliefs from a situation. Instead, you try to think about it solely in terms of the data and hard facts that you have or are able to gather.

For example, let’s apply critical thinking to a question: Why can’t women play American baseball alongside men? When I pose this question, I often hear instinctive reactions in response: “Men are stronger,” or “Men are much faster,” or “Men can throw farther” or similar. But there are no *facts* to support those statements. In fact, quite the opposite is true: there are some women who are faster than some men, some women who are stronger than some men, and some women who can throw farther than some men. If we were to think *critically* about the question, we could arrive at a theory like, “Because American society has long-held cultural biases about women’s athletic abilities compared to men.” That’s a theory which could be examined and, in time and with sufficient facts, likely could be proven or disproven. It might not be a *comfortable* theory for everyone involved in the conversation, but

critical thinking is about arriving at objective answers, not about making everyone feel comfortable.

The key to critical thinking is basing your thinking on *facts*. Tear apart whatever statements you’re making or considering, and for every element ask yourself: What *facts* do I have to support this? Where did those facts come from? Am I accepting anything as *fact*, without knowing the origin of, and basis for, that fact?

In being a critical thinker, you’ll often find yourself without the facts you might need. In those cases, it’s fine to state a *theory*. A theory is kind of like a proposed fact, or a proposed set of facts: “Given what facts I do have, I suspect that the following facts might also exist.” But in critical thinking, you can’t stop with just a theory! You have to then work to *prove* or *disprove* the theory. Once a theory has been proven, it becomes a new fact, and you can proceed.

If you’re a visual thinker, consider figure 15.1 as an illustration of a high-level critical thinking process.

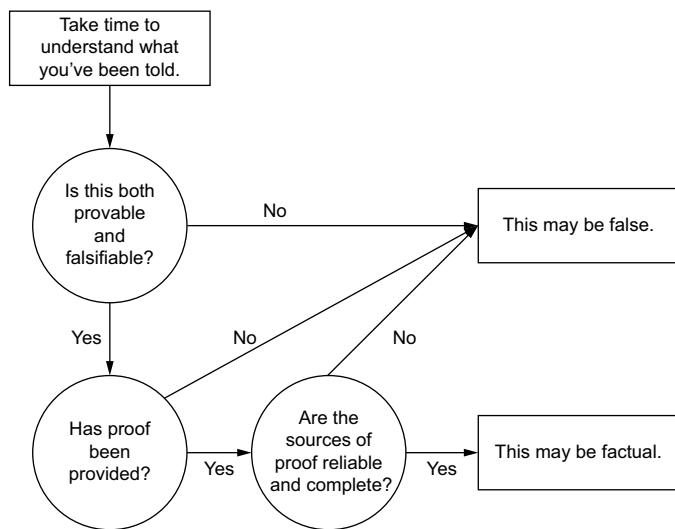


Figure 15.1 A critical thinking process

Let’s highlight the key bits from that figure:

- **Falsifiable**—If someone offers you something that can’t be proven, and/or can’t be disproven, then they’re not stating a potential fact; they’re stating an opinion. You can choose to adopt their opinion or not, of course, but don’t confuse that with facts. For example, if someone says, “Aliens never landed at Area 51 in the American desert,” that’s an opinion. It can neither be proven nor disproven by experimentation and evidence.
- **Proof**—If someone is offering a theory, then they either need to provide proof, or you need to obtain that proof on your own. Facts that are not backed up by evidence are not facts, they are opinions.

- **Reliable and complete**—This one is critical! Examine the proof that's been provided to you. Did it come from reputable sources who have nothing to gain either way? That is, are they *unbiased*? Are the sources of the proof reliable—have you heard of them before, and found their work to be accurate? And most importantly, are the proofs you're given *complete*? It's easy to "prove" almost any argument by presenting only those facts which support the desired conclusion. True critical thinkers seek out contrary evidence and weigh *everything*.

Now let's go over a couple of examples. Here's the first one: suppose you're watching a politician making a statement on TV. You happen to favor that politician—perhaps you voted for them, or you like where they stand on certain issues. The politician says, "I would be happy to provide a list of candidates for the appointment we're discussing, but the government administration has never asked me for such a list."

Many people would hear such a statement and immediately *believe* it, simply because they're positively disposed toward the politician. They might become angry at the government administration. Other people would automatically *disbelieve* the politician, perhaps because they're *not* favorably disposed toward them. All of those people are failing to think *critically*, which (I theorize) is probably where most of the world's political problems come from. A critical thinker would analyze the two statements the politician made:

- "I would be happy to provide a list of candidates."
- "The government administration has never asked me for such a list."

The first statement is an opinion, which makes it difficult to analyze. After all, who can say what might make someone happy or not? But we could still analyze it a bit. For example, has the politician said anything in the past that suggests they *wouldn't* want to provide a list? Are they contradicting themselves now? If you support the politician, then it can be difficult and uncomfortable to ask yourself that question and to analyze the statement dispassionately. After all, nobody wants to discover that someone they support has been inconsistent, or even that they might be lying. But that's what critical thinking is all about: deliberately setting aside your biases and discomfort, and focusing on what is objectively *true*.

The second statement might be easier to prove or disprove. Other media accounts or records might well indicate that the government did indeed ask the politician for a list of candidates. If so, that means the politician is simply lying. It can be extremely uncomfortable to confront the fact that someone you support is lying, but critical thinking *demands* it.

Confronted with that reality, many non-critical thinkers who supported the politician might start trying to create new "facts" to justify the erroneous statement. "Oh, the administration didn't ask *in writing*," they might offer, or, "The administration didn't ask politely." That's an attempt to preserve something as fact *which is not fact*, simply so the politician's supporters can save face. Critical thinkers avoid that behavior. Instead, if a critical thinker is *wrong*, and can be *proven by facts* to be wrong, then they simply accept it and move on.

I used a political example because most people get pretty passionate about politics; it's an area where a lot of us (me included) let our emotions get in the way of critical thinking. But we can sometimes operate from our emotions at the workplace, too! Whether it's a discussion of the "best" operating system for a particular application, or the "best" programming language for a new application, or the "best" way to engineer the network, we can all let our own biases color our statements and decisions. Critical thinking requires us to drop whatever stake we might have in the discussion, and to drop whatever outcomes we might fear. Instead, we have to focus on *facts*, and facts alone.

Now, let's look at a second example: imagine that you're at work, and you're asked to design a new user interface for an application that the company has created. One of the many choices you'll have to make might include deciding on a "dark theme" or a "light theme" for the user interface. Someone on your team says, "We should use a dark theme, because that's what everyone likes nowadays."

Do you have any *facts* to support that statement? Or is that statement really a *theory*, which needs proving or disproving? If you accept the statement as a theory, then you could conduct interviews, surveys, and other types of research to prove or disprove the theory, and then proceed with whatever new facts you uncover.

Being a critical thinker, if done improperly and without empathy, can make you unpopular. Nobody wants to work with someone who just uses facts as a blunt weapon to make other people feel bad or look bad! Here are some tips to being a *successful* critical thinker without offending your colleagues:

- Critical thinking should never be used to help you "win" a discussion. It should be a way for the *team* to win.
- You need to be a *consistently* critical thinker. This isn't a behavior you should "turn on" when it benefits you, and "turn off" when it does not.
- Consider the feelings and experiences of the other human beings in the discussion. Don't just snap, "Where are your facts?" at someone. Instead, try to guide the conversation with respect for everyone else's humanity. "That's a theory we can start with. What facts do we all have that could support that theory, or that might poke holes in it?"
- Sometimes, *there is no objectively correct answer*. A good critical thinker will recognize that: "Look, either of these programming languages would get the job done. We're split about 50/50 on which to use, and we recognize that half the team will have some learning ahead of them no matter which way we go. This doesn't appear to be a matter for facts and data at this point—just opinion. How do we want to just pick one and move forward?"
- Recognize when tightly held opinions, versus facts, may be driving a conversation. Also recognize when someone on the team may need a face-saving way to move away from the world of belief and into the world of critical thinking. When you can, be the one to help them do so, rather than beating them up about it. For example, you might try to restate their belief as a theory: "You

know, that's an interesting idea. We can certainly do some research to prove or disprove that."

- Look for the biases in what you read and hear. We all have biases; it's a natural part of being human. But when someone else is pushing an argument using *their* biases, try to recognize those, and decide if the bias is pushing a conversation or decision in a particular direction, without being backed by data. It can also sometimes be helpful for the group to acknowledge bias when it comes up: "You know, there's a possibility that's just our experience, and we probably should be careful about assuming our experience is universal." That helps put a bias front and center, and lets everyone acknowledge it and work to avoid it.

I'll offer you an example, from a fictitious news article on beef.

The ecological impact of beef

The perfect meal, for some people, is a big, juicy steak, perhaps accompanied by a fully loaded baked potato and a side of vegetables. Beef—in the form of steaks, hamburgers, and other red meat products—is as much a staple of the American diet as ice cream and Mom's apple pie. Yet the ecological impact of beef requires us to look carefully at this industry, which grew by more than 20% from the 1950s to the 1970s, according to *Beef Magazine*, a publication that covers the industry.

Experts suggest that as much as 5% of the world's greenhouse gas production comes from raising beef: cows' gaseous emissions do more than stink up the countryside; they contribute significant, dangerous gases to the atmosphere. Eliminating beef as a component of the American diet would produce a marked decrease in these gases, helping to reduce the growing effects of climate change on our planet.

These facts are bolstered by the chronic cholesterol problem in the United States and other developed countries. The US Centers for Disease Control (CDC) states that 93 million US adults age 20 or older have total cholesterol levels higher than 200 mg/dL, while nearly 29 million adult Americans have total cholesterol levels higher than 240 mg/dL (<https://www.cdc.gov/cholesterol/facts.htm>). These numbers can be significantly reduced by lowering the amount of red meat we consume.

The solution is clear: to save ourselves and our planet, Americans must move to an entirely plant-based lifestyle. Eliminating non-plant-based foods is the only way we can be healthier and the only way to stop climate change.

Such an article can be compelling, but it exhibits some biases and is not entirely data-driven. Consider:

- The 20% figure from *Beef Magazine* is accurate (https://www.beefmagazine.com/mag/beef_evolving_industry), but my article plays loose with the facts. The actual *Beef Magazine* article also states that cattle herds decreased in size beginning in the 1970s.

- The 5% number is not cited, which is suspicious. Alleging that you are presenting hard data and not citing the source of the data it is an indicator that bias may be present.
- The cholesterol numbers are cited, but red meat consumption is not the only driver of cholesterol gains. There are other contributors, including other dietary items, a sedentary lifestyle, and genetic factors.
- The article's conclusion reveals its bias. Moving to an entirely plant-based diet would also require the elimination of other animal products, such as pork, seafood, and poultry, none of which were previously mentioned in the article. It is unlikely that eliminating meat would create a reversal of climate change, even if you accept the article's uncited 5% figure. At most, the article has created a vague argument for consuming less beef.

Every day, we're confronted with messages designed to make us change our actions or change our minds. Many of those messages are **based on bias, not on data**. In your personal life, of course, you're free to follow your opinions and beliefs. In business, however, we should all strive to set aside our opinions and beliefs, and instead operate from a set of objective data. Being a critical thinker means being able to identify when bias is taking precedence over data, and to help push back toward a data-driven conversation.

15.3 **Be data-driven**

Ideally, every decision a business makes would be based upon data. That isn't always possible, though, because sometimes the necessary data simply doesn't yet exist, and you don't have the time or resources to create it. In those cases, good businesses will tend to rely on the experiences of their leaders, leaning on the past to help navigate the future. That's fine, too, but whenever possible you want to drive decisions based on *data*. Data that's objective, not subjective; that is verified, correct, and meaningful.

"Windows is better than Linux." "Java is better than C#." "Cisco is better than Juniper." These are all statements I've heard in meetings where a company's critical long-term technology decisions are being made—and none of them are based in fact. In my role as a consultant in those meetings, I try to tease out what facts exist.

For example, "Windows is better than Linux *for us*, because we have dozens of people who already understand Windows. Windows can run the Apache web server just fine, which is what our application really hinges on. We already have an enterprise agreement for support of Windows, whereas our four Linux machines don't have a formal support agreement in place." Okay—those are some *facts*, and they help lend context and meaning to the original statement that "Windows is better than Linux." We have some data points we can examine, verify, and base a decision upon.

"If we adopt this new source control system, we'll save time." That sounds like a theory or an opinion, or maybe even a belief; it doesn't sound like *fact*.

"Our current source control system burns about 4 hours a week, per developer, in overhead time while we work through code merge conflicts. Each developer is paid

\$84 an hour in fully loaded salary, which amounts to about \$336 per week or \$17,472 per year. Across all ten developers, that's \$174,720 per year in wasted time. The proposed system automates that merge process, and other users of that system said it cuts down their overhead by 50%. So we would be looking at a one-year savings of \$87,360, which far outweighs the cost of migrating and implementing the system. My opinion is that we should do a trial to verify the 50% number for our environment."

That was a data-driven set of statements. Even though one piece of data that was external and unverified, we've suggested a data-driven way forward that involves testing that 50% number and verifying it for ourselves. The payroll numbers are facts, and they're ones anyone in the company could have verified through the Payroll department. The 4-hours-per-week number could also, one presumes, be verified—perhaps by conducting a pilot project measuring labor hours more rigorously than usual. These statements have moved beyond the realm of belief: the person making the statement has stated some facts, concluded with an implied theory, and proposed that the theory be put to the test in a pilot project. *That's* the way to do it!

15.4 Beware the data

Mark Twain popularized a cautionary phrase: "There are three kinds of lies: lies, damned lies, and statistics." In other words, data can be used as much *for us* as *against* us. Statistics—one common form of data that businesses rely on—can often be stated in whatever way someone requires in order to support their opinion.

For example, suppose you're sitting in a product development meeting, and someone says, "We need to rearrange the home screen, because we have data showing that users find it confusing." They might then share a quick graph that they derived from a recent customer research project, showing that most customers do indeed find the home screen confusing. That's pretty compelling data—but it's still worth looking into.

Suppose their research was based on a survey, and the survey question was, "Do you find the home screen extremely confusing and difficult to use?" Now suppose that they surveyed just ten people. That's not very solid data: the question is written in a way that could lead someone to answer "yes" without thinking about it, and the number of people surveyed probably isn't statistically significant. A critical thinker who dug into the "customer research" might point out those shortcomings, and suggest a round of more thorough research to arrive at a stronger set of data.

Data does not collect itself. It is collected by people, or by computers that have been programmed by people. All people have biases; therefore, all data can also be biased. For example, suppose you work on a software application that has built-in mechanisms for reporting on user behavior. The collected data helps you analyze which features are used the most, amongst other details. However, you discover that—perhaps for legal reasons—your data-collection code isn't used for copies of the application deployed in the Asia region of the world. Your data is now untrustworthy, because it does not reflect the full reality of the world. The "bias" in the data may be unintentional and unavoidable, but it is a bias nonetheless.

So while it's important to be a critical thinker, and important to be data-driven, it's also important to be critical *of the data*. Make sure you understand where your data comes from, and what biases might be present in it, and how you might control for those biases before you rely on the data to drive your thinking.

15.5 Further reading

- *Critical Thinking Exercises*, <http://mng.bz/y9AG>
- *Critical Thinking*, Jonathan Haber (MIT Press Essential Knowledge Series, 2020)
- *Master Your Mind: Critical-Thinking Exercises and Activities to Boost Brain Power and Think Smarter*, Marcel Danesi (Rockridge Press, 2020)

15.6 Action items

For this chapter, I want to offer a couple of exercises to help you focus on critical thinking:

- Visit Thoughtco's *Critical Thinking Exercises* (<http://mng.bz/y9AG>) for a good, fundamental critical thinking exercise. The exercise explores the importance aspects of critical thinking through an amusing example, helping you more easily separate beliefs and biases from objective fact.
- *81 Fresh & Fun Critical-Thinking Activities*, by Laurie Rozakis (<http://mng.bz/Mg77>), is a free, downloadable book with a number of thinking activities. It's designed for kids, and I recommend you go through some of these exercises with your own kids (and if you don't have kids, perhaps ask a family member if you can "borrow" one of theirs). Watching a child go through these exercises can reveal a lot about the thinking patterns we fall into as adults, and help you take a fresh look at your own thinking.

Understand how businesses work

Most of us work in a business or business-like environment; even nonprofits and many government agencies have a predominantly business-like approach to the way they work. It therefore makes sense that you should understand the “rules of the game” for businesses in general. Doing so can help you contribute more effectively, and navigate your career journey more efficiently.

16.1 Businesses are people too

In most countries, businesses are, from a legal sense, considered *entities* in the same way that people are. Companies have individual rights and obligations, often pay taxes, can own and sell property, and so on. The Romans of the mid-500s (mid-500s *A.D.*, that is) recognized a range of corporate entities; the word *corporation* itself comes from the Latin *corpus*, meaning a *body*, or more specifically in this case a body of people.

I want to take the comparison between *business* and *person* even deeper. Companies, like people, have needs. They have motivations, which often relate to those needs. Getting judgmental about other peoples’ needs is easy. For example, I cannot comprehend why the people in the condo above mine needed the yappiest little dog as a present this past Christmas, but here we are—they had a need, they filled it, and I don’t understand it. Anyone or anything with different needs or motivations than yours can be difficult to understand, and even easy to look down on, but it’s important to acknowledge that we all are different. Other people have needs that differ from your own, or may even be in opposition to yours. That is just how life works.

The same is true for businesses. Some people look down on businesses because “All they want to do is make money.” True, but that’s more or less the precise reason

you start a business, so nobody should be surprised when that's what the business' primary motivator turns out to be. Of course, it's worth noting that plenty of businesses do a lot more than *just* make money: many offer significant support to their communities, for example. Every business, like every person, has different motivations.

Like people, businesses engage in relationships. Like any relationship, everyone in the relationship is expecting to get something from it, and the relationship only works well if everyone is getting at least *most* of what they need. One-sided relationships aren't healthy, are often frustrating, and are the ones most likely to end in a fight. So let's talk about businesses and their relationships.

16.1.1 Businesses and their relationships

Probably the first kind of business relationship that springs to mind is the one businesses have with their customers. That's an easy one, right? Customers get whatever it is the company sells, and the business receives money, and everyone's basically satisfied with the relationship. There are degrees of "satisfaction," though. I filled the car up with gas the other day: the gas pump worked, and the price . . . well, it was what it was. We don't have that much price variability in my neighborhood, so I pay what I pay. So I suppose both myself and the business were satisfied, albeit on a somewhat vague, done-and-done level. I mean, I'm not explicitly planning a return engagement, and I don't dream wistfully of those minutes spent at the pump. It's a *good* relationship that we have, but not a great one. It's nothing like the relationship I have with a little restaurant called 7th & Carson, just a couple of blocks from my condo building.

I *love* going to "7th&," as we call it around the neighborhood. The food is terrific, and I feel it's an excellent value for the money. I look forward to the charred octopus in particular, although the chicken wings are pretty spectacular as well. I like the staff—Oscar behind the bar who always makes such great drinks, the chef who not only turns out excellent food but is also fun to chat with, and the owner, Liam, who's got a delightful Irish accent and always makes us feel welcome. As a customer, I think I have a great relationship with 7th&. I enjoy telling people about that relationship, as you can see.

Like people, if not more so, businesses *need* relationships in their lives. A company with zero relationships isn't going to go to therapy or stay home and play video games; that business is going to *die*. Businesses not only need relationships, they need *good* ones. Many companies go out of their way to try to foster good relations, even when those relationships get a little one-sided in favor of the customer (like the saying, "the customer is always right"). Businesses will often stay in toxic relationships a little longer than they should, making you wonder if there's some sort of couples counseling for companies and their customers. For example, I've seen really abusive customers return to a store again and again, constantly buying and returning merchandise, and I wonder why the company doesn't just cut them off and save all the hassle.

But customers aren't the only party that businesses have relationships with. Businesses also have relationships with their vendors. They may have relationships with

organizations like the news media, the local government (for licensing and inspections), and so on. However, the *most* significant relationship a business has is with its employees.

16.1.2 Customers and employees

You might argue that the customer is king, but I'd phrase it a little differently. A business *is* its employees; the relationship a business has with its employees is absolutely required for the company to, by definition, exist in the first place. A business with nobody doing any work is just an idea, not a functional entity. True, businesses are more tolerant of toxic relationships when it comes to customers, but that doesn't make the employee relationship less critical.

I view the customer relationship a bit like meeting the relatives of your significant other. You *know* there's going to be one or two weirdos in the group, and you're prepared to tolerate them. Most of the rest are fine, and if you're lucky, there's a handful of really brilliant characters in the mix.

The employee relationship is a *lot* more like the friends you grew up with and have stayed in touch with your whole life. You all know way too many dirty secrets about one another, you've seen each other drunk too many times, and you're a little too quick to upset each other sometimes. But you're also comfortable with each other, which means you're generally more tolerant when you've upset each other, and you tend to come back to each other. That doesn't mean you're "best friends forever," though: even the best of friends can have a relationship go toxic, and while it's always painful and dramatic, sometimes the relationship has to end. That's how it is between businesses and employees, too.

Also, like any relationship, the quality of the business-employee relationship can fall on a spectrum. In the *best* relationships, both sides are getting what they need, they're happy about it, and all's well in the world. In the *worst* relationships, neither side is happy about it, nobody is having their needs met, and everyone else wonders why they don't just call it off.

We hear the word "entitlement" a lot these days, and it's really easy to point at someone who expects things to be a certain way and call them "entitled." I'm going to choose to not use that word because I think it's a little overloaded with sentiment. Instead, I'll go back to that relationship analogy.

16.1.3 One-sided relationships

When I was just out of high school, I had a small group of friends who hung out quite a bit, and it was pretty common for us to wind up at a diner or burger joint somewhere in the evenings. Two of my friends stand out in my memory: one, whom I'll call Jon, was always really up-front when he couldn't afford to eat out with us. "I can't, guys, I'm short until my next check," he'd say. We'd all nod, and either we'd offer to cover his tab, or he'd go off on his own if none of us could. Often, one of us would go with him and do something else, or we'd all call off the eating-out altogether. Another, who

shall be known as Shelly, would invariably wait until everyone had ordered, finished their food, and the check had come, before announcing she had no money. Shelly was annoying that way. I mean, we *liked* Shelly. She was fun, and she didn't mind sharing popcorn at a movie, presuming she had the money to pay for the movie and the popcorn, but she'd always wait until the rest of us were basically on the hook before announcing that she wasn't participating. After she'd eaten, of course.

Shelly was an example of a bad relationship. She was clearly getting something from the relationship, but the rest of us weren't getting what *we* needed, which was some mutual respect and not being taken advantage of. If our group was the "business," then **Shelly was a lousy employee**.

I run into many employees who are like Shelly. They seem to feel that a job is something everyone is supposed to have and that so long as they're showing up most of the time, and putting in minimal effort, then they deserve to get paid. I don't think of this as *entitlement*, although, again, that's the word you hear a lot. I think of it as a sad, one-sided relationship.

I also run into many employees who are like Jon. They seem to feel that if they're not interested in doing whatever the job is, that they should leave and go work somewhere else. You can imagine which type of employee I have more respect for. Again, though, this is just a different place on the relationship spectrum, a place most of us would regard as healthy and desirable.

Jon and Shelly are, of course, extremes. Most of us are neither Jon nor Shelly; we're somewhere in between. But if you were to think about your relationship with your employer, how would you characterize the relationship? And no, don't think about how *you* feel about the *business*; give yourself an honest appraisal of how the business might feel about *you*. If you were in charge of "couples therapy" for your relationship with your employer, what observation would you make? What advice would you offer?

If it's easier, modify the relationship analogy and consider the contractual relationship between you and your employer. You may actually have an employment contract, but if you don't, pretend that you do. What does the contract require of your employer? What does it *really* require? If it's silent on the length of a work week, for example, then you've no reason to presume it will be 30 hours or 60 hours; it "is what it is." What does the contract require of *you*? Presumably, your job description outlines your duties. If you accepted the contract, explicitly or implicitly, then you need to hold up your end of it. The business needs to hold up its end, too—believe me, I've seen plenty of businesses that were at the toxic end of the relationship, and I'm not trying to make employees out to be the Universal Enemy. But I am saying that it takes two to tango; if you feel you signed a bad contract, or that the other party isn't upholding its side of the bargain, then point that out. Bring it up! Discuss it, and ask for change. If the change isn't forthcoming, and it's important to you, then you can end the relationship.

A friend of mine, Bob, was in a particular job for about six years. He was hired to do software quality analysis, which is a fancy way of saying he watched a bunch of

machines run automated tests against a bunch of other machines to make sure the second set of machines did what they were supposed to do. When they didn't, he sent test reports back to the machines' programmers, who made fixes so that Bob could do it all over again the next day. Bob got bored of this after about six months, which is completely understandable. (Smart companies have automated all of this these days, so they do not need a Bob at all anymore.) But Bob stuck with it. Sort of. Initially, he began looking at the broken code and making suggestions for fixes. The programmers were delighted by this because Bob was doing their job for them, out of his sheer boredom. Bob asked for a pay raise and was turned down. Maybe not a great call on the company's part, because Bob was going above and beyond, right? But also maybe a good call, because all the company wanted Bob to do was what they were paying him to do. Everything he did "above and beyond" was fine, but nobody had asked him to. And that's when the relationship went a little toxic, and it's honestly when Bob should have started looking for another job elsewhere. But he stayed on for more than five additional years.

What'd he do in all that time? He caused trouble. He'd start rejecting code not because it failed its tests but because he'd spot stylistic errors. Basically, he'd reject functional code because he didn't like the color shirt it was wearing, which wasn't one of the criteria he'd been given in the first place. After getting yelled at about it, he started just *fixing* the stylistic errors he saw. Of course, these would sometimes create *functional* problems, causing the code to fail in the field—because Bob had passed it, so it was presumed to be working!—which would get the *programmers* yelled at. All the while, Bob would just remark how lucky the company was to have him, doing all this extra unpaid work.

Except the company *didn't want him to do it*. They'd had an implied contract for Bob's job and what he would be paid. That had never changed. Well, it had: *Bob* had changed it. He'd unilaterally renegotiated his contract, reinvented his job, and then gotten bitter that the other party to the contract wasn't on board. From a relationship perspective, the company didn't *need* this extra fuss Bob was causing, and eventually, the relationship got toxic enough that Bob was fired. Bob was that friend who moved in for a weekend and stayed for a month: "Dude, I *like* you and all, but this is *not* what I wanted the relationship to be."

The funny thing is, some folks could argue that *either Bob or the company* was the original proximate cause of the problem. I mean, sure, the company didn't want to pay Bob more to do a job they hadn't hired him for—weird, right?!? Bob could have just gone back to doing what the relationship initially called for, but he didn't. I argued with him and said, "You know, Bob, it's like you moved in with a roommate, agreed that you'd both sleep in separate rooms, and then he finds you in *his* bed with him every morning. That's gonna upset him. You changed the rules of engagement without everyone buying in." Bob *firmly* felt the relationship's souring was the business' fault. And that's how relationships work, right? Once they go wrong, nobody wants to own responsibility. So responsibility almost doesn't matter—bad is bad, and sometimes the best you can do is end it before it gets *worse*.

16.1.4 Dealing with changes in the relationship

I'm not suggesting you quit your job. I *am* suggesting that you have a job because a business needed a skill or service that you could provide. Maybe they needed some programming done, or a network fixed, or some servers maintained. But the business had that *need*, and you came along and offered to do it. *You* presumably needed money and benefits and what-have-you, and you both agreed it was a fair exchange. *The moment that situation changed*, it was on one of you to say something. “I’m bored of programming,” you might say. “Do you have any other needs?” “No,” the business might have answered. “It’s programming or the highway,” at which point *you’re the one who asked for the change in the relationship*, so you need to decide if you can continue as it was or if you need to move on. Also, it can go the other way: “Hey, employee, I no longer need programming done. That is not a need of mine anymore. I’ve changed, and I need you to maintain the servers now.” “Well, business,” you might say, “I love programming and that’s what I signed up for and what I want to do.” The relationship changed. It would be sad if you had to leave because of a reason like that, but if you’re no longer *satisfying each others’ needs*, then the relationship can’t continue without becoming toxic.

Does that make any sense? Businesses are like people, and people change. The needs a business has today might not exist in ten years. The things you’re willing to provide to a business might not stay the same for five years. At some point, one of you will have *different needs*. *That’s nobody’s fault, provided someone can speak up, acknowledge that the relationship no longer works, and see if there’s a new relationship that does. Where it gets toxic is when you know the relationship no longer works, but you want to keep plugging along and ignoring it. Just as in your personal relationships, that is never a good idea, and it’s what makes everything eventually end badly.*

As you engage with the business, look at your relationship with it as well as at how others relate to the company. You’re absolutely going to run across companies who can’t maintain a healthy relationship with *anyone*. That’s a shame, and the bigger shame is when the people in that relationship aren’t able to recognize it for what it is, or when they’re trapped in some way and not able to get out of it. We’ve all heard stories about people who were in bad personal relationships that, for whatever reason, they didn’t feel they could get out of.

If you find yourself running a business, be mindful of that relationship. Employees aren’t “resources” to be utilized to their maximum potential and then discarded; your business is in a *relationship* with them. Sure, you need something from that relationship, but they do as well.

I think businesses go wrong when the people running the business stop thinking of the company as a person, and when they stop realizing that there’s a true *relationship* between the business and its customers, vendors, and employees. I think employees tend to go wrong when they forget that, too. Always approach relationship problems from the angle of “what could make this relationship better for *both* of us,” and see if that helps create a better situation or at least creates some clarity on whether the relationship *can* be good or bad.

16.2 How businesses really make money

When you work for a company, it's really, really, really important to understand what the company *actually* does for a living, and how the company itself—or those with a financial stake in the company, like its investors—measure the company's success. Gross revenue is rarely the most compelling metric about a business, and for many modern businesses, gross profit isn't even the most important metric. I explain *gross profit* and *gross revenue* in Chapter 19, by the way.

Consider the story of MedVidCo, a company I've invented for this story. MedVidCo is in the business of making videos about medical procedures, and they sell their services on a subscription basis to doctors. Doctors pay over \$10,000 a year for access to MedVidCo, and the company employs some of the most well-known doctors in the world to create their videos. Their library serves as an invaluable reference to doctors, who use the videos to learn new techniques, brush up on ones they've not performed in a while, and so on. MedVidCo has been successful enough that they've attracted several rounds of private investment to fuel growth, and they've just recently completed an Initial Public Offering, or IPO, to become a publicly traded company. Think of MedVidCo as “the Netflix of medical videos,” where you pay a flat fee for unlimited online access to stream anything from their entire library.

Joey is a salesperson for MedVidCo. He recently attended a medical conference and met one of the experts who creates videos for the company. They had a great afternoon chatting about how well the company was doing, and the expert brought up a point Joey had never considered. “Why,” the expert asked, “don’t you guys have us also create videos for ordinary people? You already have us experts, and we could cover topics like general wellness, explaining various conditions people deal with, and so on.”

Joey was intrigued but pointed out that ordinary people weren’t going to pay \$10,000 a year for access to that kind of video.

“Sure,” the expert said, “but you’d sell it to them for like \$100 a year instead. So you’d make less per customer, but you’d have tons more customers. Think of how much the revenue would go up!”

Joey took the idea back to the office after the conference and created quite a conversation within the Sales team. However, when they finally took the idea to their executives, the executives killed the idea almost without discussion. Joey was depressed. He couldn’t understand why a sensible company would turn their back on potentially millions in revenue and started wondering if he was at the right company. To be dismissed so categorically was disheartening, and it seemed like the executives just didn’t “get it.”

The problem here is that Joey doesn’t have all the facts about how his company makes money. He is unaware of three facts in particular.

First, no form of revenue is ever free. You always have to spend money to make money. In this instance, a new line of business would require new marketing campaigns, which cost money. You also have to worry about how long an ordinary person

would maintain a subscription; If you spend \$80 per subscriber in marketing, and make \$100 for the first year, and then the person doesn't renew, then you only cleared about \$20. That's not very good.

Second, no resource is infinite. While your video-making experts might be able to make content of interest to ordinary humans, if they did so, they'd have to temporarily *stop* doing content for your high-paying doctors. At \$20 profit per ordinary-human per year, you'd need 500 of them to make up for one doctor subscription. Yet while your experts were churning out ordinary-human videos, you might start *losing* doctor subscriptions because you're not turning out the quantity of doctor-level content they're used to. This is called an *opportunity cost*, and it's something we'll discuss later in the book. It's the cost you incur when you do Thing A instead of Thing B.

Third and perhaps more important, Joey doesn't know a thing about how the public market values MedVidCo. It turns out that for their kind of subscription-based company, the two key metrics that tell the market how well the company is doing are Average Subscriber Revenue and Subscriber Retention Rate. With subscribers paying \$10,000 and renewing regularly, the company looks fantastic. Throw in \$100 subscriptions, though, and that Average Subscriber Revenue goes *way* down. Moreover, if those ordinary humans don't renew as reliably as doctors, then the Subscriber Retention Rate goes down. All of a sudden, it looks like MedVidCo isn't a healthy company and is instead doing fire-sale pricing to try to shore itself up. People start selling the stock, driving its price down, and making it harder for MedVidCo to borrow more money to fund future growth.

Now, sure, Joey's executives could have explained all of that, and in a good company, they'd have done so. But if Joey were really a *businessperson*, he'd have *asked* those questions up front. He wouldn't have assumed that raw revenue or subscriber counts were the main things everyone worried about; he'd have *asked* what the company's key metrics were.

My experience with most executives (although certainly not all) is that they have no problem answering questions about how the company works, how it is valued, or what metrics they rely on to run the company. Most are *eager* to share that information if they can do so. What tends to rub them the wrong way, though, is people coming up to them with The Next Great Idea without bothering to ask any questions up front. Imagine being inundated all day, every day, with "great ideas" that fundamentally don't fit how the company works, all brought by people who never took a minute to try to understand if their ideas were, in fact, "great" or not. You'd get irritable, too, and you might even stop paying attention to the genuinely good ideas that sneak through. That would be unfortunate, but it's human nature to a degree.

Whether you're trying to pitch a "great idea" or not, though, you should take an interest in understanding why your company exists, what motivates it, and what it considers "success":

- What problem does our company attempt to solve in the marketplace?
- What are the key metrics we use to determine overall company health?

- What resources do we use to run the business, and what are their inherent limitations?
- Who are our customers? What do *they* see our company doing for them?

I almost added a fifth item to that list: “Who are our competitors, and what do they do differently?” That’s a question you *should* be asking, but it’s a loaded one. Once you know the answer, that doesn’t mean your company needs to try to do the same things as your competitors. You need to be very cautious about correlation: “Well, our competitors do *this* differently than us, and they’re making a ton more revenue than we are.”

Joey ran across the same thing. MedVidCo’s biggest competitor is HumanVideos. HV does a similar line of business, but they ship physical Blu-Ray discs rather than streaming videos. HV’s customers pay a one-time fee for each package of discs rather than a subscription fee. HV is also publicly traded, so it’s public knowledge that they make a lot more revenue than MedVidCo does. It was reasonable, then, for Joey to suggest that MedVidCo should also sell Blu-Rays to customers as an alternative to the streaming video library.

However, upon deeper digging, it turns out that HV’s *market valuation*—that is, what the public market feels the company is worth—is about what HV’s current revenue is. In market terms, HV is valued at 1x, or “one times,” their earnings. That’s not much, and it means the market doesn’t see a lot of growth potential. In other words, you wouldn’t invest much in HV, because they’re never going to earn much more than they’re currently worth.

MedVidCo, however, is valued at three times its current revenue, which means the market sees a lot of room for MedVidCo to get bigger. You see, HV could go out of business at almost any time because they rely on continuously selling Blu-Rays to new doctors, and more Blu-Rays to the same doctors. They don’t have any guaranteed recurring revenue; each new sale is a one-and-done thing. MedVidCo, in contrast, has recurring revenue. Even if a doctor goes on vacation for a month and isn’t watching videos, MedVidCo still took in their subscription fee. MedVidCo has to worry about renewals, but that’s often easier than booking a whole new sale. MedVidCo’s subscription model is part of what makes their valuation so high, and it’s something they’d lose if they added a Blu-Ray sales option.

Businesses can be *complicated*. They involve customer psychology and market forces, and they can involve truly subtle and complex measurements and considerations. You should try to understand as many of those as you can. Doing so not only makes it easier for you to have a healthy “personal relationship” with the company but can line you up to make better decisions, which will eventually line you up to take on more responsibilities, including—if it interests you—leadership. *Not* understanding the hidden details of the business means a lot of what the business does will seem inexplicable or even stupid, and that’s a terrible way to feel about a personal relationship that’s as important as the one you have with the company that employs you.

16.3 What does your business sell?

Do you know what your business sells? Consider the following three stories. While you're reading, try to imagine what the company thinks it does for a living, and where there might be some room for improvement in its business models.

Now, obviously, these are fake companies, and obviously, I'm lining you up with trick questions. Try, for a moment, to forget that they're trick questions. Put yourself into the day-to-day life of Terri, Martin, and Pat, whom you'll meet in the following three sections. Try to see things from their perspectives. What, if you were them and they wound up in control of their companies, would you change?

Obviously, a problem with all three of these scenarios is that you won't have the full download on what these companies are all about; you're only getting a biased view from one employee. However, that's how *most* employees operate, and it isn't always *entirely* their fault. Many companies do a poor job of helping their entire group of employees genuinely understand what the company is all about. Often, that's just because the company's leaders haven't thought to do so, and if someone asks, they're usually happy to share. Let's dive in.

16.3.1 Example 1: Terri's International Bulbs

Terri works for International Bulbs. The company stocks an enormous range of light bulbs, from old-school fluorescent tubes to leading-edge color-changing LEDs. She's employed as a programmer, working mainly on the computer systems that control the company's warehousing and distribution center operations. In working on those systems, she's noticed that the company tends to stock only minimal quantities of most bulbs. She knows from looking at the data that the company sells plenty of bulbs, but many times an order will come in that can't be immediately fulfilled. Instead, there's a delay of several days until the warehouse receives the stock and then ships it out to the customer.

What Terri doesn't realize is that International Bulbs makes its money as a *subscription service*. Subscriptions are what make the stock market love the company because they have incredibly predictable revenues, and most of the company's sales and marketing efforts can go toward winning *new* business rather than having to continually "re-win" business from existing customers.

The company's warehouse works precisely as designed, which is to minimize the amount of back-stock they keep on hand. Back-stock is expensive: In most countries, under Cost of Goods Sold (COGS) rules, you can't write off the cost of the goods you sell until you've sold them. So if you buy \$100,000 worth of light bulbs, you *pay income tax on that money* initially, and only recoup the tax when you sell the bulbs. So you tend to try as hard as possible *not* to keep much back-stock.

Instead, Terri's company does just-in-time (JIT) stocking. For example, if they know that a customer is on a quarterly subscription for a bulb, and they know it takes

them a week to get that bulb in stock, then they can simply work the dates backward so that they order the bulbs, receive them, and then ship them to the final customer “just in time.” The customer doesn’t see any of that happening on the back-end. What Terri sees in the warehouse isn’t an “out of stock” situation; instead, it’s the stock being received precisely when it’s needed so that it can then be shipped out to the final customers right on time.

The question Terri might instead be asking is why International Bulbs doesn’t engage in *drop-shipping*, a practice where the company places bulb orders with its suppliers, but those suppliers ship directly to the final customer. There are times when that might not work, such as when International has to buy a massive case of bulbs and then break down that case to ship to multiple customers, but drop-shipping—when it’s practical—would be a way to save money by cutting out International’s “middle-man” warehouse.

16.3.2 Example 2: Martin’s Theme Parks

Martin works for Global Themed Amusements (GTA), a company that owns regional themed amusement parks across the globe. Martin works in the Purchasing department, primarily focused on negotiating deals for maintenance supplies that are used throughout the company’s properties. Martin has recently been concerned about what GTA’s competitors are doing. While the competition is building ever-more-thrilling rides like multimillion-dollar roller coasters, the GTA parks tend to expand more slowly, and tend to install less expensive “dark rides.” The company also spends way more on retail merchandise than its competitors do—and the shops used to sell that merchandise often take up valuable space that could have been used for rides and other attractions.

This example is about understanding *what your company sells* and recognizing that it might not be what everyone “commonly” thinks they sell. In the case of GTA, Martin’s worried because the company is being overtaken in the thrill-ride department by competitors. However, GTA as a company doesn’t see themselves as being in the thrill-ride business at all. They see themselves as selling a *family experience*. They focus on attractions that families can experience together and that create strong and positive memories. One way they profit from those memories is by selling merchandise that reinforces the rides’ themes and imagery. Retail merchandise at the parks earns a strong 70% profit margin, which winds up paying for a much larger percentage of new ride construction than the parks’ admission fee accounts for on its own.

See, just because you see another company doing something *similar* to yours — building theme parks, for example—doesn’t mean they’re actual competitors. Good companies try to *differentiate* themselves from the competition so that consumers don’t have a simple A-or-B decision to make. If you’re selling a commodity like gasoline, consumers pretty much make a decision based on factors like price and convenience,

which is why most towns have so many gas stations—they’re all fighting to be more convenient for some chunk of the population. It’s also why gas companies spend so much time trying to *differentiate* on the quality of their gas, or their rewards programs, or whatever. It pays to *really* understand how your company believes it differentiates itself.

16.3.3 Example 3: Pat’s Fruity Clothing

Pat works as a salesperson for Fruity, a high-end clothing brand that markets primarily to teenagers. Fruity doesn’t sell directly to consumers; instead, Pat’s job is to work with buyers from major online and brick-and-mortar retailers. Fruity’s clothing is pretty expensive—most items run 3–4 times as expensive as similar items from cheaper competitors. Pat gets frustrated that so many lower-end retailers won’t even consider stocking Fruity, and feels the company’s price point excludes them from many markets. That means Pat doesn’t make as much in commissions as would otherwise be possible, which is where the frustration originates.

Many businesses differentiate themselves through *perception*. Take Apple, which sells phones, computers, and other electronics that routinely cost hundreds more than their competitors’ products. How do they “get away” with it? By creating a *brand* that certain consumers want. In any given market, you’ll always have consumers who are price-sensitive and will go for the least expensive version of a thing that is available. You’ll also have consumers who are brand-sensitive and will go for the brand that they *perceive* as being the most valuable. None of those consumers are wrong; they’re just being met in the marketplace by different vendors.

This thinking is why car companies like Toyota, Honda, and Nissan have separate brands like Lexus, Acura, and Infinity. It’s why some people buy Rolex and others buy Timex. In the case of Pat’s company, selling into “lower end” retailers might indeed open up more commissions for Pat; it might also erode the “high end” brand that the company has worked to achieve.

I’ve worked for companies that play this “branding” game, and it can be tough. I’m mainly a price-sensitive consumer with things like clothing, and so for me, selling a pair of jeans for \$110 seems silly when I can see other brands selling for \$20. However, companies in those spaces deal with a far more complex range of issues than you might think. While both pairs of jeans probably cost a similar amount to make, the \$110 pair is creating a much bigger profit margin, meaning the company doesn’t have to sell as many pairs. Moreover, consumers who are brand-sensitive will tend to buy more than just jeans: they’ll also buy shirts, shoes, and accessories, and they’ll often come back several times a year to do so. There’s an entire brand loyalty scheme at play, which usually doesn’t draw the price-sensitive consumer as much. Again, none of these companies are *wrong*—they’re just operating in different spaces, for various reasons, and with different risk/reward situations.

Sure, Fruity could charge less and be in a broader array of retailers, but then they'd be a *different company*. It's unfair to ask a business to be someone different. After all, they were who they were when *you* got there, so if you want them to be someone else, then maybe *you* should be looking for a new line of work, instead?

As a quick aside, there *are* clothing companies that attempt to address every market, but they tend to do it with different brands. Consider Banana Republic, Gap, and Old Navy: three different brands with different styles and price points, but all owned by the same company.

16.3.4 **Know the details of the business**

The point of all this is to *know why your company makes money* and to know *how* they make money. Sure, there are likely other ways your company could operate. Those other ways aren't wrong, but what your company is currently doing isn't necessarily wrong, either. Ask questions that help you understand *why* your company behaves the way it does.

I'll offer you a story that was told to me when I started asking those questions. "Why," I asked, "is our fiscal year starting in February and not January?" Our CEO, a long-time operator from the department store days, laid out a calendar.

"You get four seasons in a year," he said. "Each season is three months. The first month of each season is where you introduce the season and often make the most profit by selling new merchandise at full price. The middle month is your primary sales for those seasons and you tend to move the most units then, although often at slight discounts. Last month is your clearance."

"The first season is Spring clothing—February, March, and April, lined up with Easter roughly in the middle. Next is Summer clothing, May, June, and July, with the end of school right in the middle. Next is Fall, August, September, October, with back to school in the middle. Last is Christmas covering November, December, January."

"The reason we want January in there, and on the same fiscal year as Christmas, is so that your returns come in on the same fiscal year as the sales. Otherwise, because Christmas often has so many returns, you start your fiscal off in the negative, and that looks bad."

It was an eye-opening story, and it explained a lot, all of a sudden, about the retail cycles I'd been working in for years without ever realizing it.

16.4 **Understanding risk and reward**

Risk is a fundamental concept in business that people don't talk about enough. In any venture, any kind whatsoever, there is a risk of failure. Just driving to the grocery store involves risks: someone will hit your car, the store won't have what you need, your credit card will be declined, or whatever. Risk is all around us. For the small, day-to-day risks, most of us have learned to mitigate what we can automatically: we drive carefully, we pay our credit card bill on time, and so on. However, most of the risk that most of us deal with on a day-to-day basis is *within our control*. We decide to drive carefully, we

choose to check the store inventory online before we leave the house, *we* decide to pay the credit card bill on time. We can't eliminate the risks, of course, but we can, through personal action, mitigate those risks to a pretty large degree.

Businesses are a little different. To start at the end of the conversation, the risks faced by businesses are rarely within the businesses' control. Instead, a business relies on other people, its employees, to mitigate the risks. Think about how mentally challenging it would be if that concept applied to your personal life! Imagine that you're planning a big family vacation, and some stranger is handling absolutely every last detail, and they're not even really sharing those details with you. That'd make *me* a nervous wreck, although if I'm honest, I run across people on vacations all the time who do it just that way. They're usually miserable because they couldn't take any personal responsibility for their enjoyment, and their vacation planner forgot some crucial detail, and it all went wrong at some point.

However, I digress. People can *choose* to outsource their risk mitigation or handle it personally, depending on how much of a control freak they are. Businesses don't have that choice. They *always* rely on other people, their employees, to deal with every possible risk.

So what risks are we talking about? Predominantly financial. Every business is, at any moment, just a few bad decisions away from losing everything, going bankrupt, laying off all their employees, and disappearing from existence altogether. The *owners* of the business are specifically the ones at risk because they're the ones who put up all the money in the first place. That's why the owners of the company get to share in the business' success, although *great* businesses will, through bonus and other types of programs, try to share some of that success with the employees who helped make it happen.

Actually, bonuses are a good way to think about risk and reward. Suppose you work for a company that has a gain-sharing program, or profit-sharing program, or something similar. When the company makes a profit, the owners agree to set aside, say, 25% of that profit, and divide it up amongst the employees. Cool, right? At the end of the year, or quarter, or whatever, every employee gets a nice chunk of extra change as a way of thanking them for helping make the business a success. However, what if the company *didn't* make a profit? What if it *lost* money? Would you, as an employee, expect to have money withheld from your paycheck? Almost nobody would, but it would be fair, right? If the business succeeds, you get a chunk of the profits; if it fails, shouldn't you pay a chunk of the losses? Say the company lost \$100,000—shouldn't the owner take 25% of that, or \$25,000, and divide it up as a deduction from everyone's paychecks? After all, if it's the employees who make or break the business, and if they share in the successes, wouldn't they have responsibility for, and a share in, the losses?

I'm not aware of profit-sharing programs that do that, which is why they're fundamentally not a way of sharing the risk in the business. See, when you have a proper share of risk, then you succeed when the company does, and you lose when the company does. *Risk* means you stand to lose something. The stock market is all about risk:

When the companies you invest in fail, your stock value goes down, and you've lost money. That's why stockholders are literally considered co-owners of their companies, and it's why they get a say in how those companies are run, often by appointing a Board of Directors, who in turn, hires the executives who run the company daily.

Risk and a say in things go together. If you have no risk—that is, there's no chance of you losing money—then you don't get much of a say in how things are run. This is a basic rule every parent teaches their children, whether they realize it or not: "You live under my roof, you follow my rules." As the parent, you've got all the risk, right? You have to hold down a job, make sure bills get paid on time, watch out for your kids' health, and so on. All the risk, so you get to make the rules. Well, at work, most employees are "living" under their employers' "roof," and so the employer—the one shouldering all the risk for running the business—gets to make the rules.

I've spoken with *many* employees who don't understand why the business doesn't just do whatever the employee thinks they should do. "It'd be so much better!" they say. Nearly universally, those employees lack context about how the business makes its money, and nearly universally, they lack any actual share of risk in the business. I understand that it's frustrating to work in an environment that won't "listen to you," but how much did your parents listen to you about proper television allowances when they were paying all the bills? However, when you finally moved out and got your own place, you could stay up and watch as much TV as you liked, right? Same thing in business: If you want a say, then *you* take the risk. Start your own business. Walk away from the sure-thing paycheck, the benefits, the 401(k), the free coffee, or whatever. Instead, worry about where your next check is going to come from. Worry about whether your employees are going to make the right decisions that day and keep the company running. Put up the money, take the risk, and you get to make all the decisions.

Another reason I find that companies get a little deaf when it comes to listening to employees telling them how to fix everything and make it better is that most employees lack sufficient context. They can't see all the moving parts that make up a company. Not that the company is hiding anything; it's just that businesses, especially large ones, are horrifically complex. Take the company I currently work for: We have to file income taxes in something like two dozen US states and a half-dozen foreign countries. That simple fact creates a level of business complexity I can barely comprehend. Literally every day-to-day company decision gets affected by that fact. "Should we rent some new office space so that we're not so crowded?" becomes an epic question when you think about different states' tax incentives, the depreciation rules on office build-outs, whether the office is located in a city where we've agreed to employ a certain number of people in order to get certain incentives, whether the lease comes out as a simple expense or not—it's mind-numbing. You'd think it'd be a simple answer, like, "Well, if the lease is \$3,000 a month and we're currently making more profit than that, then sure!" but it isn't even *close* to that simple. That's why when I hear employees (not at my company, mind you) complain about petty things like whether the kitchen has the coffee K-Cup flavor they prefer, I start laughing so hard I almost cry. The day-to-day

brainpower that goes into the seemingly simplest company decisions is so much that “coffee flavors” likely didn’t even make it on anyone’s agenda. Coffee flavor is something you can only complain about from the comfort of a nothing-at-risk position. The people managing the actual risks of the company, the ones charged with keeping it in business and making sure our paycheck gets cut on time? They’re probably delighted that the water is still running to the building, and less concerned about what flavor coffee people are making with it.

So. You want to have a louder voice in the running of your company? You want to be *heard* by the people in charge?

Start by making sure your voice is more than just noise. Learn what the business is *for*. Learn why it exists and how it makes money. Then learn how companies are run. That doesn’t mean getting a Masters of Business Administration (although that wouldn’t hurt), but you’re going to need to learn about business finances, business management, and a lot of other topics. You can do it—the people running your company didn’t drop out of the womb with that knowledge, right? They learned it somewhere. You can, too. Then look for ways to put some skin in the game. People will take you more seriously when you’ve something at risk. So if that’s not possible at your current company, ask yourself if you’re ready to go to a different one or start your own.

Context. Business acumen. Risk. Those are what create a voice in business.

16.5 Further reading

- *Understanding Business* 12th ed., William Nickels, James McHugh, and Susan McHugh (McGraw-Hill Education, 2018)
- *The Personal MBA: Master the Art of Business*, Josh Kaufman (Portfolio, 2012)

16.6 Action items

For this chapter, I’m going to ask you to look at some of the details of your business, whether that’s your own or a company you work for. If you don’t know the answers, find them! Start with your own manager, but ask if it’s okay to contact others in the organization for their perspective as well:

- What does your business really *sell*? What is the business model, and how do the people with a financial stake in the company measure the business’ success?
- What sorts of risks does your business routinely deal with—and what is your part in those risks? Are *you* at risk of anything other than possibly losing your employment if the business can’t afford to keep you at some point?
- What are your business’ motivations? What sorts of things does the company measure to track its performance? Why does the company behave, at a high level, the way it does? Do you agree with its reasonings, or not?



Be a better decision-maker

Decision-making is all around us, almost every day. Being able to make better decisions, in almost any circumstances, is a real career-booster. The key to better decision-making is to understand the basics of how you can evaluate a situation, and how decisions themselves are made—particularly in business settings. Another key point of this chapter is that it's crucial to get all the different bits of context if you hope to comprehend the decisions that your company makes.

17.1 Deciding who decides: Decision-making frameworks

The worst kinds of companies are the ones where a small number of leaders, each with a strong personality, make seemingly arbitrary decisions and then inflict those on the rest of the business. Employees don't understand the thinking behind the decisions, have no input into the decisions, and often feel like they're just cogs in a machine. It's not a very rewarding feeling.

The *other* worst kinds of organizations are the ones who invite everyone to be a part of the decision-making process, but don't provide any means of resolving conflicting input, conflicting priorities, or any other conflicts. Those companies often get stuck in analysis paralysis, going round and round with what they could do, discussing the pros and cons of every possible choice, and in the end making no choice whatsoever. The best organizations have a decision-making framework in place, so they have guidelines for how they want to make decisions.

To use an extreme example, most military organizations have these decision-making frameworks built right in. A general may say, "We need to take that hill in order to gain a tactical advantage." But the general knows perfectly well that it's the

sergeants who know how to take hills and that the sergeants' expertise is an important input into the decision-making process.

"You know, this other hill is a bit higher, and positioned just as well tactically," the sergeant might offer. "It'll also be a little easier to take, since it's already closer to our lines."

The general might nod and reply, "Yes, but our allies are going to take that hill. I know it's easier, but it's been a rough ride with them for the past few weeks and we need to give them a win if we're going to maintain a civil relationship with them." The sergeant would likely nod and head back to their team to start planning.

The military is built around the idea of lower-level experts providing input to higher-up decision makers, and then accepting the decision. I just found out today, as I was sitting down to work on this chapter, that Amazon has adopted a similar leadership principle, which they call, "Have Backbone; Disagree and Commit," meaning that you can and should offer your input, but if the ultimate decision-maker goes a different direction, get behind that decision and do your best to make it happen successfully.

My current employer uses a decision-making framework called RAPID, which was developed by Bain, a management consulting company. RAPID stands for Recommend, Agree, Perform, Input, Decide, and for most major ongoing decisions, we identify the job roles within the company that own each part of the RAPID, so it's clear to everyone who is involved with each of these five steps. Here's how we break down these roles and responsibilities:

- *Recommend*—Makes a recommendation for the decision; in the case where there is no clear-cut "win," may make multiple recommendations and outline the pros and cons of each. For example, a senior developer might *recommend* a particular software library for use within a project.
- *Agree*—Stands behind the recommendation, which means they'll work with the Recommend role to come up with a recommendation everyone can *Agree* with. For example, given a software library recommendation, the DevOps, Security, and overall Development teams might need to *agree* that the library will work for their various needs and concerns.
- *Perform*—Executes the decision; usually provides valuable input to the recommender on what will and will not work. For example, the actual software developers using a new software library will *perform* the implementation of that library.
- *Input*—Provides additional input and data to inform the recommendation; this includes business analysts and other decision-support roles. For example, a variety of people potentially impacted by the adoption of a new software library might have *input* into which library is chosen.
- *Decide*—A *single person* who ultimately decides what will be done, and is accountable for it. For example, the manager of software development might ultimately *decide* on which library to adopt, based largely on the *recommendation* brought to them.

ORDERING RAPID BY CHRONOLOGY RAPID is a nice, easy-to-pronounce acronym, but it confused me when I first learned of it because the tasks aren't performed in the order in which the acronym implies. Chronologically, you'd start with the Input, get people to Agree to a Recommendation, Decide what to do, and then Perform. But IARDP is a terrible acronym!

Figure 17.1 shows the chronology for how a RAPID decision is often made.

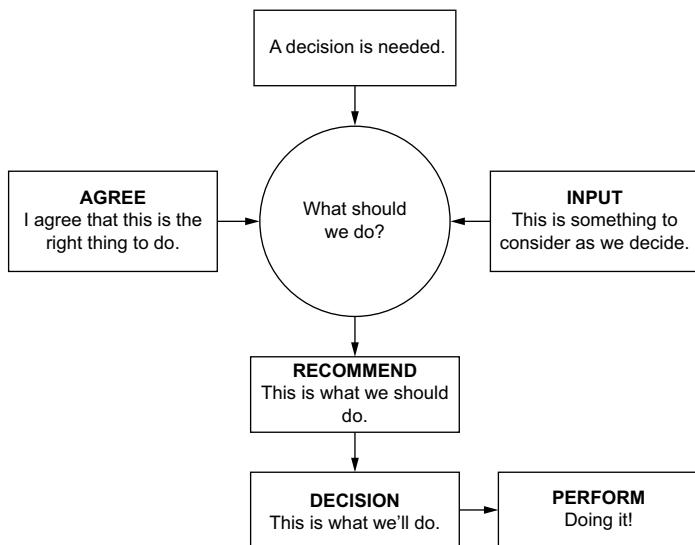


Figure 17.1 Making a RAPID decision

Every bit of RAPID except the “D” can be a single person or a group of people. The fact that only one person can “own the D” means that there is one voice, typically someone responsible to the business for whatever outcomes are being considered. This is the person who ends the debate and adopts the direction in which everyone will move. Once the “owner of the D” makes that decision, it’s everyone else’s job to implement it. The decision doesn’t have to be popular, and it doesn’t have to be something everyone agrees to, but it is the decision, and everyone must work to put it in place.

CAN ONLY ONE PERSON “OWN THE D?” It’s possible for a small group of people to “co-own the D,” but it can create problems when that group can’t unanimously come to the same decision, right? After all, if they “own the D,” who acts as referee? Sometimes, it means reconsidering ownership. For example, in one organization I’ve worked for, we had certain high-level decisions that my team’s executive technically “owned the D” on. However, for day-to-day purposes, her leadership team of four people had *delegated* ownership. Pro-

vided they all were on the same page, the executive didn't get involved and those four "co-owned the D." If they couldn't come to a consensus, she would step in and exercise actual ownership.

As an individual contributor, your job is to do the best you can at providing *data-driven* input, helping the decision-maker make the most informed decision possible, and compromising when needed to create agreement. Once the decision is made, it's your job to get on board, and not continue to argue or debate. This framework only works when everyone can commit to their roles, and commit to respecting the final decision that's made. There are plenty of other decision-making frameworks out there; the point is that a good company has one in place, so everyone at the company knows how decisions are made.

17.2 Deciding what to do: OKRs, rocks, and pebbles

Almost all companies have more things they could be doing than they have resources to actually do. That means they need to decide which things to focus on. Given all of the things we could do, in other words, which ones will create the most positive impact? Deciding what to do, and what not to do, is a huge deal.

Frameworks like RAPID can help identify the people contributing to, and making, those decisions about what to do and what not to do, but there are additional models that businesses can use to help weigh the pros and cons, as part of the decision-making process. These models can also help the entire company understand why a particular prioritization decision was made, and help keep the company focused on those priorities that the company had decided to pursue.

These models can help on anything from a small team to an entire business, and they're a great way to communicate what the priorities are, and to keep everyone aligned to those for a given period of time (like a quarter, half-year, or year).

17.2.1 Rocks and pebbles

The first is an analogy from Stephen Covey, who wrote the best-selling *The Seven Habits of Highly-Effective People*: rocks and pebbles. Imagine you have a bowl. This bowl represents the time you have in a day, or month, or quarter, or whatever. The bowl is, therefore, a fixed size: just as you cannot create more time out of thin air, you cannot make the bowl any bigger.

Beside the bowl are rocks of various sizes. Big ones the size of your fist, and little ones you'd need tweezers to pick up. These are all the things your company could choose to do. The big ones take up the most time and require the most resources, and the little ones take the least time. As a result, the big ones also tend to be the ones that generate the most significant impact on the company, and the little ones tend to generate the least impact. A big rock might be creating a new customer self-service portal, where customers can look up answers to questions about your products; a little pebble might be picking up the phone and answering a single customer's question.

Decision-making frameworks like RAPID can be used to decide what's a rock and what's a pebble, giving every appropriate role in the company the right amount of input, and ensuring the decision sits with whomever will be ultimately accountable for the outcomes that the decision is meant to drive.

Prioritization becomes a process of deciding what to put into the bowl. You *could* just fill it with little pebbles. They're easy, and they get you an "instant win." They're often the "low-hanging fruit" you hear people talking about. But they're not going to make a definite long-term impact, even taken as a group. You *could* just focus on the big rocks that you know will create a significant long-term impact—but doing so will mean ignoring some of the little, day-to-day tasks. It goes without saying that you can't do it *all* because the bowl is a fixed size. The trick is in deciding what mix to put into the bowl. And the *big* trick is realizing that *there's no objectively correct answer*, and the *biggest* trick is realizing that *you won't know if you got it right until the bowl is full and you can look back*.

This kind of prioritization is a struggle for all people, and all businesses, all of the time, always. You've always got more that you want to do, and not enough time/money/ whatever to do it. You've always got people happy to second-guess your decisions, but you don't know *in advance* if their decisions would have turned out any better in the end. Running a business is a *compromise*; it involves making a lot of bets over the short, medium, and long term, and it's often a long time before you see if those bets paid off or busted. Any time you're thinking, "I don't know why the business doesn't just do *X*," it's because adding something to the bowl would necessitate taking something else out. And while *X* might be some wee little project that wouldn't take any time at all, *X* might also be a pebble that won't generate the kind of long-term impact that the company needs and wants. In the end, you might be right—doing *X* might have been smarter than whatever else was being done. But if you're not the person with skin in the game (see chapter 15's discussion on risk), then you're not the one who gets to make bets with what goes into the company's bowl.

17.2.2 OKRs

A related and sometimes complementary model is Objectives and Key Results (OKRs). An Objective is something you want to gain by performing some set of actions. Relating back to the rocks and pebbles analogy, an Objective is the reason you put the rock into the bowl. Objectives have to be observable, which means an objective person needs to be able to look at the situation and determine whether or not the Objective happened. "Increase Sales" is certainly an objective, and it's something that can be measured and agreed upon. "Increase customer loyalty" might be an Objective, if and only if you have a plan for accurately measuring customer loyalty in the first place.

Key Results, or KRs, are the ways you tell whether you've achieved an Objective. "Customer Satisfaction Surveys Exceed 80% Completion with a Rating of 75% or Higher" is a good Key Result.

“Conduct Customer Satisfaction Surveys” is *not* a Key Result; it’s a task that you might undertake on the way to achieving a key result. “Conduct 100 Surveys” is also not a Key Result, or shouldn’t be, unless you’ve defined some “soft” Objective like “Learn More About What Our Customers Think of Us.”

I don’t like soft Objectives because they don’t move the business in any direction, and they aren’t tied to a meaningful business outcome. In most cases, I prefer Objectives that specify something the business’ customers would care about.

So let’s consider “Learn More About What Our Customers Think of Us.” Do customers care if our company learns more? Probably not. And just because we’ve learned more doesn’t imply that we’re *doing* anything with that new knowledge. Simply performing tasks doesn’t necessarily *achieve* something that will create a positive impact on the company. So how might we improve that? By asking *why*. Why do we want to learn more about what our customers think of us? Perhaps it’s because we want to increase customer retention. In that case, a better Objective would be, “Increase Customer Retention by [some useful measurement].” One means by which we might achieve that is better understanding what our customers think of us, and so we might establish a Key Result similar to “Survey [some meaningful number] Customers to Understand Why They Would or Would Not do Business With Us Again.”

Notably, that Objective will clearly create a positive impact on the business. That Key Result is a measurable step toward creating that impact.

Objectives are things that will measurably improve the business; Key Results are the measurable milestones along the way that let you know you’re headed in the right direction.

OKRs are something a company can set for itself, and it’s good when it can do so. But you can also do OKRs at a department level or even within small teams. Ideally, team OKRs will be connected to a department OKR that they support, which will be connected to a company OKR that it supports. The idea here is to get everyone doing the tasks that will help the company achieve its overall goals. Teams and departments will often have other OKRs that don’t directly address a company OKR, but are still relevant and worthwhile.

17.2.3 Priorities, priorities

Clearly I’m a fan of companies (and departments and teams) that use mechanisms like these to communicate top-level priorities to the whole company. However, I realize that not all companies do this. But there’s no reason you can’t ask. Ask how you, in whatever your position is, can understand on a day-to-day basis where the company, department, and team priorities lie. Ask how those are measured. Ask whomever makes sense, likely starting with your direct leader and working up the org chart as needed. Make it clear that you want to line up your own daily actions with whatever will have the best impact on those priorities. If you’ve got suggestions, lining them up against those priorities will be the best way to have them heard. “Hey—I know our priority is to get new customers to sign up faster. I had an idea about modifying our sign-up

screens that would shave about 10 seconds off per customer. Where would be the right place to take that suggestion?"

17.3 Deciding what to drop: Opportunity cost

In our physics-based universe, almost nothing is limitless. Every resource we have is finite, and that means we have to make choices about how to use them. If you get two weeks of vacation from work, you have to decide what to do with those—much as you might want to take an 80-day trip around the world, you can't, unless you're willing to lose your job. Businesses work under the same constraints: to do something is to also *not do* some other thing. This is *always* true. I'll sometimes get amused by employees who'll make observations like, "Well, we *could* do this thing that I think should be done, and it's not like it would cost more or take any extra time," when those same employees are usually the ones also remarking on how overworked they are and how little time they have to do anything extra. Every endeavor will require some expenditure of resources, and that means those resources won't be available for some other project.

Businesses usually discuss this decision-making process in terms of *opportunity cost*. If I do thing A, what will I miss out on? I've had to struggle with this question myself innumerable times in the various businesses I've owned or run over the years. Mind you, it's a lot easier to wrap your head around this concept for a small-scale business because the numbers involved are both smaller and more readily available. For example, we once had an opportunity to do some extraordinarily exclusive and lucrative consulting work for a division of Microsoft. However, with our sharply limited resources, it meant basically dedicating the entire company to that work and nothing else for about a year. That dedication meant rapidly closing out our existing commitments, taking no new commitments for that period, and, more seriously, potentially damaging our pipeline for future work. By stepping away from our then-current line of business for that long period, we risked those customers throwing up their hands and finding someone else to do what we'd been doing, which means after the year-long gig was up, we'd potentially struggle for new work. The *opportunity* with Microsoft, in other words, came with a *cost* that we'd have to be willing to pay.

Number-crunching commenced. So, to use big, round numbers, let's say the company was making \$500k a year doing what we were doing. Let's say we figured it'd be six months to re-ramp back to that after concluding the consulting gig, with revenue slowly coming back online during that ramp period. It means our "opportunity cost" was about \$675,000, meaning the Microsoft gig needed to pay that much or it'd end up losing us money.

Opportunity cost happens everywhere, all the time in business because to do something always means to not do something else. It's just a numeric way of looking at the rocks and pebbles analogy; if my bowl is only a certain size, and I need to decide which rocks to put in it, I need to look carefully at the value of those rocks. I need to select the rocks that will produce the most long-term value because the bowl is of a fixed

capacity. With everything you do, you have to learn to ask yourself, “What value am I creating in doing this, and what potential value am I forced to leave behind because I’m doing this, instead of something else?” Ask that for every project, every meeting, every initiative, at some point down to every task you take on and every moment of your day.

Now, look: if you’re not in a leadership role in your company, you might not get to make the decisions about what gets done and what, as a result, doesn’t get done. However, you should ask questions, and work to understand the decision-making process that went into those decisions. You should try to get as much context as possible around things like opportunity costs so that the business decisions are comprehensible to you.

Let’s take a moment to talk about that word *comprehensible*. I learned, as I was coming up through various companies and figuring out how businesses were run, that I needed to work to comprehend a company’s business decision before I formed an opinion on it. There’s a fine line between arrogance and confidence. Confidence is knowing what you know, and arrogance is not knowing what you don’t know but behaving as if you do. I don’t need to agree with business decisions until after I’m sure I fully comprehend the decision and where it came from. In fact, I tried very hard to assume that if I saw a decision I disagreed with, I was probably wrong, and I’d work as hard as possible to understand why. At the end of the day, there were still plenty of decisions I didn’t agree with. When those started to pile up at a given company, I started brushing up the résumé. But more often I found that given a decision I didn’t agree with, if I learned a bit more, I could understand and accept it. My acceptance might be tempered with thoughts like “Oh, OK. I mean, I wish there was another way, but I kind of get it.” But at least I understood why it came about.

Business decisions are often a matter of choosing the least-destructive option from a long-view perspective, and while you don’t always have to love the decisions, you can often at least respect the company’s intent behind the decisions.

Opportunity cost is one of the important pieces of context that go into business decisions. It’s far from the only one, of course, and different types of business have different concerns. The point here is that there are different bits of context, and you need to understand as many of them as possible if you hope to comprehend the decisions that your company makes. Work with your leader to not only gain that context, but to have them validate your correct understanding of the context you’ve gained.

17.4 Deciding what’s enough: Good, better, best

Back in the day, one of my responsibilities was for our company’s PBX—the phone system. Our system was relatively new, and it was well-equipped, but one particular area of challenge was the voicemail system. We had many salespeople who spent much time away from the office, and they saved *every* voicemail. Our system kept filling up. We’d run out of room for the entire company, and callers couldn’t leave voicemails. I should mention that this was before internet email was broadly available, so voicemail

was crucial. My boss asked me to contact the vendor and work out an expansion so that the company could look at the budget. So I sat down and did the math. I'd already been going in every time the system hit "full" and manually clearing out older voicemails. So I had a good idea of how fast we were going through our storage and what it would take to hold about 30 days' of voicemail companywide. We needed to roughly double our capacity.

Capacity expansions for that system came in "blocks," and I could either add a "small block," which would add about 20% or a "big block," which would add about 400%. There was no middle ground. I ran up a quote for a big block and took it to my boss. This solution was, I reasoned, what we needed and then some; in my young, "engineering mindset," I'd rather overbuild and not run into the problem again, right? It was a lot of money—close to a quarter-million if I remember correctly. Phone systems are expensive. My boss pointed out the massive expense and asked if there was anything else we could do, and I said—truthfully from my perspective—"No." I mean, a small block wasn't going to make any appreciable difference.

A few weeks later, I had another meeting with my boss, and she chastised me. *She* had called the phone system vendor and spoken to them in more detail, and they'd said that it was entirely possible to install multiple "small block" expansions and that we could get a roughly 80% increase—close to the doubling I'd been after—for much closer to \$170k, a bit over half of what I'd been quoted. It turns out the "big block" also came with a bunch of other stuff we did not need, and involved some prerequisites we did not have, and both these contributed to making it so expensive. I learned two lessons.

Lesson one: ask more questions. I had made some assumptions that turned out to not be correct. For instance, I assumed that, for our very large company, \$250,000 was not an insurmountable sum. The money meant nothing to me because I had no idea what the money meant to the company. I had no idea what else we might need money for. So I didn't dig very deeply into the possible solutions. I didn't take the time to really understand that "big block" option because I didn't see the need to. It was just money, right? We had lots of money, right? I was perfectly happy to throw someone else's money at a problem because I had no context for that much money, and what it meant for our company. Asking more questions helps you correct or validate your assumptions, learn more about the business, and make better recommendations and decisions. Ask your direct leader first, of course, but try to form relationships with other leaders in the company so that you can gain more context for different situations.

Lesson two: offer options. I also learned this lesson from a friend who works in Procurement at a large resort hotel. In Procurement, you try to reduce the number of things you're buying so that you can get better deals by buying fewer types of things in greater quantities. There were several restaurants on the property, and each chef was very particular about their tools and supplies. Every chef specified different . . . well, everything, down to spatulas. With ten food outlets, he'd be buying ten different

kinds of spatulas. So, when he took over, he put a stop to this. “I have three options,” he said. “There’s a good one, a better one, and the best one, in terms of price and quality. The low-end outlets like the buffet are assigned the good one, because it’s less expensive and perfectly adequate for their needs. Higher-end outlets can pick from the three choices, since we afford them a little more latitude on their expenses.”

In expanding the voicemail system, I should have researched it to the point where I could provide a good, better, best scenario. Then, in my meeting with my boss, I could say, “We can do this option for this much money, but it won’t meet our needs for very long. We can do a better option, and it’ll come closer, but it’ll cost a bit more. We can do the best option, which will set us up for life, but it’s going to cost a good bit more.” Then my boss could have evaluated those options and chosen one, asked me to dig into them more, or modified the criteria a bit and asked me to go back and reprice options that fit the new criteria..

By only offering one option, and settling on that option with essentially no business context, I was not helping my boss; I was compounding her problem. My “solution” didn’t fit with all the business criteria she was juggling but that I was unaware of. With a good, better, and best option, I could have provided her with some context, with a sense of the *shape* of the solution. “Here are three ways to tackle this, and the trade-offs between them,” I could have said. This is what we needed to inform the next round of more-precise questions that we could have used to drill down to an actual solution.

Now, to be fair, my boss could have helped me realize all that at the time. So in a way we were both wrong in different ways.

I’ve since learned to present good, better, best options along with a concise summary of the trade-offs, a recommendation about how I think we should proceed, and the assumptions that went into that recommendation. Those assumptions are essential because my boss can quickly scan them to see if there are any considerations I did *not* take into account and determine if I missed some parameter that might change the equation. By implicitly letting my boss see where I’m missing information, I create a better opportunity for my boss to fill me in on the missing pieces. This process educates me more about the company’s needs and priorities and helps me make better recommendations the next time around.

Business leaders are not always asking for solutions. Sometimes they’re asking for options, because they’re trying to inform themselves about the shape of a problem. They may ask from a particular perspective, or they may ask questions that don’t immediately seem relevant, but that’s because they know how they themselves process and learn from information. My boss and I need to be partners in the decision-making process. The boss is charged with a particular set of responsibilities and outcomes that I’m not entirely filled-in on because it’s not *my* job to know all those things at all times. So I need to provide a spectrum of options that give my boss a feel for the overall situation and the possible trade-offs between good, better, best. My boss, in turn, can help me better understand their concerns, their outcomes, and what they’re accountable for, which enables me to bring recommendations that are better aligned with those.

Very notably, I try *not* to “steer” my boss in one direction or another. If I believe I have a recommendation that is a best-fit for the situation, I’ll say so, and I’ll try to explain why I believe that. But I’m *always* cognizant that I might not have all the information. I may be missing context, or I may be unaware of some criteria that my boss knows. As my boss fills me in on additional context, I’ll revise my recommendations accordingly.

17.5 Deciding what to believe: Being data-driven

Human beings are unique on earth in our ability to *believe*. That is, we can accept and treat something as a fact even if there is absolutely no evidence for it, or even if there is evidence that contradicts what we’ve accepted as fact. This is different from merely holding an opinion or stating a theory; we behave as if the thing we believe is absolutely, irrefutably true, and are often resistant to a discussion about how our belief relates to any available physical evidence.

Your dog doesn’t get excited around dinner time because it believes you’re going to feed it; it gets excited because it’s operating from a pattern of past activity where you did in fact feed it. That’s not a guarantee that you’ll do so again, but a dog’s world is based on those past experiences, which definitely occurred.

A human, however, can believe, for example, that a particular politician committed a particular crime or other offense, with no experience or objective evidence to back it up, and can continue believing that even if an overwhelming amount of evidence refutes their belief. We want a thing to be true, and so we create a narrative for ourselves where it is true and then behave as if our narrative has been confirmed by objective, physical evidence.

This way of believing gets especially tricky in business. Good businesses do not operate from belief; they run from theory, and they confirm or refute theories based on facts. That is to say, good businesses are data-driven. So if you’d like to be taken seriously in such a business, you too will need to become a data-driven person.

I used to work for a division of the telephone company Bell Atlantic (it’s now part of Verizon), and at one point I was responsible for migrating us from our aging and overwhelmed Lotus cc:Mail messaging system to a new system. Much of the rest of Bell Atlantic was going with Lotus Notes, another IBM product and the “obvious” successor to cc:Mail. As a standalone division, however, we were free to make our own choice. So my team and I started collecting data. We had some specific criteria. We needed, for example, to be able to place a fixed maximum size for each user’s mailbox. Our research discovered that Notes didn’t provide that capability; you could set a maximum size on an entire mail database, but a database would typically contain many, many different mailboxes. One colleague said that his company’s workaround was to have one database per user, effectively giving them a per-mailbox size limit, but with the expense of vastly more complex system management, backups, and performance problems. We made a note of that, along with a variety of other facts about our business needs, Notes’ capabilities, and the capabilities of Microsoft Exchange Server, the primary competitor to Notes at the

time. In the end, we expressed the opinion that, based on the data, Exchange was the better choice for our division's business needs. This choice was not a *belief* of ours; we had collected objective facts that informed our recommendation. Some of my team were die-hard Notes enthusiasts; others loved Exchange. However, we agreed that the facts would be the primary driver of the decision.

Our recommendation was not popular with some of our superiors who *believed* Notes was just a better product. However, our division's leadership was accustomed to making data-driven decisions. We brought up our fact-chart. We went through each business criterion and asked our superiors to confirm that we had each one correct and that each was still relevant to the discussion. We indicated how each product met, or didn't meet, each criterion. They nodded, asked a few good questions, and accepted our recommendation because it was data-driven and the facts that drove our recommendation were valid, meaningful, and relevant.

I'll point out that we took some pain to ensure our facts were free of any bias that couldn't be supported by more facts. While facts themselves are, by definition, true, the way you present those facts can indeed impart bias and create, if not a false impression, then at least a steered impression. We tried to avoid that because when someone catches you trying to "spin" a fact, they start to call the rest of your facts into question, and the whole process impugns your credibility. Neither Notes nor Exchange came off as perfect in our analysis; both had problems that we'd need to work around. We tried to provide context for those workarounds, to categorize their difficulty and long-term impact, their costs, and so on, and we tried to be very clear about where we departed from objective fact and entered a world of estimates and guesses. We were, to sum up the entire process and attitude in a single word, *scientific*.

Many people have difficulty pulling themselves away from what they want to be true and operating instead in a purely data-driven fashion.

For example, I have one friend who believes that Android phones are superior in every way to iPhones. That's an opinion, and he's certainly welcome to it, but there are no facts that support *either* platform being objectively better than the other. Instead, he will take pieces of information—not data *per se*—and "spin" those to support his opinion: Android is more open, he'll say, which means it is inherently more secure. Android is less expensive, he'll say, which means more people will buy them which will make them part of a larger and therefore more robust ecosystem. He is an extremely vexing person to be around when he starts on that path, and you wonder exactly what kind of financial incentives, and from whom, he's under to be such a die-hard proponent. *Belief* is his problem. For most humans, it is not enough to believe something on your own; you must also win others to your belief—as if belief was a kind of democracy where the largest crowd of believers "wins" and gets to have their beliefs magically become proven fact.

Try not to be that person at work.

People also mistake *correlation* with fact. "IBM servers," a former colleague once told me, "are far less stable. I can prove it because we have to reboot them 3-4 times as

often as our Dell servers.” I pointed out that the IBM servers were only used to run a particular application that was known to be poorly written, and that it could very well be the application, not the server hardware, that necessitated the frequent reboots. The fact that the crashes *happened on IBM hardware* was a *correlation*—two distinct things that happened at the same time—rather than a *causation*, where the one thing (being an IBM server) automatically led to the other (frequent reboots). By his logic, we could state that tomatoes are toxic to humans because every human who has ever eaten a tomato is either dead or will be dead at some point. There’s a correlation—people do eat tomatoes, and people do indeed die—but the two are not causally connected. This argument got to be such a stalemate that I installed the poorly-written application on a Dell server, which wound up needing to be rebooted just as often, and removed it from an IBM server, which suddenly didn’t need to be rebooted as much. Because my colleague so committed to his belief, however, his response was that I’d messed up the servers somehow—the evidence itself didn’t sway him.

Try not to be that person at work, either.

Here’s the thing: for millennia, humans haven’t had data. Sure, we had experience, but experiences can be subjective, and they’re not always universally shared. So learning came, in large part, from instinct and perceived, as well as actual, cause-and-effect relationships. So even in today’s modern, always-connected world, it’s still easy just to observe and then go with your gut. That’s a terrible way to become a better businessperson.

I can already hear the objections, so let’s be clear: observing and going with your gut is a great way to form a theory about something. “Hey, I see this happening and I think this is what I should do about it.” Fine. But then gather data. Gathering data is rarely fun (unless it’s something you’re personally into), and it’s not always easy (which is why data engineering is such a “thing” in businesses these days). However, it’s what you need to do to make a sound decision. Dig into the data around that thing you see happening—do you see something whole and real or just the edge of something or something completely imaginary? Let the data show you how, your gut-instinct solution will help—or how it won’t help. So if you decide to go ahead with your solution, measure more data. Consider yourself a scientist: “I think this is what’s happening, and I have some data to validate it, so I’m going to try an experiment. I’ll measure that experiment, and decide if I made a positive change or not.” Be *that* person at work.

17.6 Deciding together: How to negotiate

Negotiation is a hugely important part of *any* relationship, and the many relationships of the business world are no exception. For example, negotiation comes into play whenever you are involved in

- Figuring out what your team can do and what they’ll need to drop
- Getting hired or promoted and discussing your compensation package
- Resolving an interpersonal or interteam conflict

Negotiation is a way of making decisions *together*. Even if you’re using a decision-making framework like RAPID (which I discussed in the first part of this chapter), negotiating is often one of the first things that happens. The various people in the Agree role are *all* supposed to agree on the recommendation (or recommendations) that are being put forward; that almost always involves some negotiation between them. “Yes, my team can commit to that—but only if your team can commit to picking up this other responsibility we currently have.” The final recommendation, which is sent to the ultimate decision-maker, normally includes all of those negotiated trade-offs, so the responsible business leader can make a decision.

Negotiating can get tense and nerve-wracking sometimes, because it can get so personal: I don’t want to give up whatever it is you’re asking for, and you don’t understand why that’s so important to me! I’ve created a kind of checklist for myself, to help me navigate negotiation. My list might not be appropriate for the type of salesperson-customer negotiations that you normally think of when you see the word “negotiate,” but I’ve found it useful for internal negotiations where we’re all basically on the same team:

- *Have I been clear about my priorities, what lines I don’t feel I can cross, and why I feel that way?* In other words, am I sharing context with my fellow negotiators, so they know I’m not simply being obstinate? This is an opportunity to validate my priorities and “lines not to be crossed” as well, to make sure they’re still supporting the organization’s overall outcomes.
- *Is everyone putting something on the table?* In a healthy negotiation, you should get something for every thing you give. That’s not intended to mean that negotiation is a zero-sum game and everyone has to both win and lose; it’s a recognition that it’s only a negotiation if everyone is putting something on the table. If I’m the only one being expected to “give,” then it’s railroading, not negotiating. Sometimes that’s how the business has to operate, which is fine; I just want that fact on the table up front so I know what’s happening.
- *Have I taken the time to understand the contexts of my fellow negotiators?* Do I understand why they’re asking for things, and how those things relate to the organization’s top-level outcomes? It might be a situation where they’re right in asking my team and I to do something, in which case the “negotiation” is just us figuring out what might need to be dropped in order to make it happen.

I’ll offer a personal example: I once worked for a startup that was having trouble securing its next round of funding, and was burning through cash too quickly. I was a senior leader, and was being paid a considerable salary. They came to me and asked me to take a salary cut. Obviously, that didn’t thrill me, but I understood why they were asking: it was in part to ensure they could continue to pay other people too, and I wasn’t the only one being asked. I decided to negotiate. I asked if, in lieu of my full salary, I could be granted additional equity in the company that vested monthly. In other words, I asked if I could be paid partly in cash, and partly in company stock. It demonstrated my belief that the company would eventually succeed (or the stock

would have been worthless), my willingness to contribute to the greater good, and my understanding of the situation. But I added another request: I wanted *twice* the value in stock as the salary I was giving up. That put the company on notice that I wasn't going to be the only one making a sacrifice in the situation—I was giving something up, and they would have to as well. After some discussion, we agreed.

Negotiation can be a healthy part of any business. Most businesses try to create intentional “points of tension” within the org chart, with different stakeholders—like Finance or Legal—representing different perspectives. There are deliberate “tensions” between them. For example, Finance might ideally prefer to spend no money at all, while Marketing might want to spend everything! They come together and negotiate, and between them come up with solutions that best represent the competing concerns that any business has to have. When that negotiating can be done professionally and respectfully, and with an eye toward the shared outcomes of the entire organization, then it's a great way to mutually create the right decisions.

17.7 Further reading

- Bain's RAPID framework, <http://mng.bz/aKgz>
- *Decision Quality: Value Creation from Better Business Decisions*, Carl Spetzler et al. (Wiley, 2016)
- *Measure What Matters: How Google, Bono, and the Gates Foundation Rock the World with OKRs*, John Doerr (Portfolio, 2018)
- *Start Less, Finish More: Building Strategic Agility with Objectives and Key Results*, Dan Montgomery (Agile Strategies Press, 2018)
- *Business Analytics: The Science of Data-Driven Decision Making*, U. Kumar (Wiley, 2017)

17.8 Action items

For this chapter, consider some of the decision-making that happens in your own organization. If you don't know the answer to these, find out! Start with your own manager, and work together until you both know the answers:

- Does your company use a decision-making framework like RAPID? If not, is there value in considering such a framework for the commonly made decisions on your team?
- How does your company communicate priorities? When upper management communicates priorities, do the successively lower levels of management “translate” those into more-specific actions for their teams?
- How does your team or company deal with evaluating opportunity cost? That is, if an opportunity comes along that would require dropping something else, how are the tradeoffs evaluated and how is a final decision made?
- What data sources does your team or company use to make decisions? Where does the source data come from, and how is that data made available to decision-makers?

Help others

I will forever argue that one of the most valuable ways to measure your own life and success is to count the lives you have helped to succeed. When I think of *professional brand*, I come up with few attributes more positive or impactful than “helps others.” I believe that being able to help others, and doing so, is one of the most powerful acts we can do for each other as humans, both personally and professionally.

18.1 Why help?

In my view, teaching is the ultimate career level-up. If you look around you, the technologists who are doing the best in their careers are probably the ones who share what they know with their colleagues, with their technical community, and with their peers. And honestly, being willing and able to help other people—to *teach*—is simply the right thing to do. You have likely been helped, and will be helped, by those around you; you owe it to them to “pay it forward” and help others.

Whether you’re teaching formal classes, holding a “lunch and learn” at work, or just leaning over a colleague’s shoulder and helping them with a problem, your effectiveness as a helper—as a *teacher*—is something you can continually improve. With that in mind, this chapter is intended to help you boost your skills as a teacher, and as someone who can act as a force multiplier for the technologists around you.

And before you start thinking, “But I don’t have anything to teach,” let me stop you. You *do*. You may just be stuck in the toxic loop of thinking that “teaching” only happens in a classroom, or in a training video, or some other formal setting. That’s not true: teaching and learning happen all the time. Leaning over someone’s shoulder to help them solve a problem at work is *teaching*.

And you have plenty to offer. Maybe you're just stuck looking "up" at the people you admire and respect, and thinking, "What could I possibly teach them?" Stop looking "up" for a moment, and look "down" and "around" instead. There are plenty of people in the world who know less than you; find them and offer them your help.

18.2 Yes, you can

Before we dig into the main part of this chapter, I want to go a bit deeper into that feeling of not being qualified to teach, because people frequently tell me things like, "I'd love to teach other people, but I don't really have anything to teach," or "I'm not good enough yet to really teach others." Let's get this false and restricting belief out of the way.

18.2.1 The toxic relationships that keep us from teaching

We have two toxic relationships, which have for the most part been given to us by our culture and society, that hold us back from truly helping others. The first is education. When we're growing up, most of us attend school. Many of us continue attending school—in the form of college or university—after our primary education is done. For that time period, education is something that happens in a special set-aside time, in a special set-aside place: during school, and in a classroom. For the most part, our teachers were set above us as figures of authority. They didn't want us to stand up and try and teach anything, and our ability to participate in the class was limited to those times where they asked us to.

At no point did most of us ever get a certificate of some kind that said, "You're ready now! You can start teaching other people! Congratulations!" That means we enter adulthood, and our working lives, with some serious misconceptions:

- Learning happens in special set-aside places called classrooms.
- Learning happens only during special set-aside time.
- You have to be a specially trained person in order to teach.

Worse, there's an overlapping toxic relationship that plays into this: our relationship with our role models. These are the people in our field that we look up to, and it's frequently far too easy to compare ourselves to them. "If I don't know as much as they do," we reason with ourselves, "then I'm not good enough to teach other people. After all, *I learned from my role models!*" All these misconceptions hold us back from doing what may be the best thing we can do in our lives: *help others*.

18.2.2 You are definitely worthy of teaching

We all need to stop looking "up" to our role models, measuring ourselves against them, and judging our ability to help others based on our role models' abilities. Instead, we need to look "back" at the people who are less experienced than ourselves. Those people might exist within our workplace: junior developers, systems administrators just starting out, or entry-level network engineers. But they might also exist

outside our workplace. We're surrounded by people who know less than we do, and who could help lift themselves up, and do more for themselves, with our help. They live in our own community, and in the communities next door to ours.

Teaching and learning is not an activity that only occurs in classrooms, and it isn't something that can only be done by specially designated teachers and students. We all learn, all the time: the problem you solved by querying Google was *learning*. That family member you showed how to configure the mobile app for their new robot vacuum? You were *teaching*. We need to recognize that learning and teaching are *always* happening, and to be more deliberate about taking the time to do it.

You *are* worthy of teaching. If you ever think you're not . . . you're wrong.

18.3 How humans learn

Learning is largely a function of memory. That is, when we experience something, and perhaps make a mistake, we form memories about that event and what we did. The human brain is wired to dislike failure (<http://mng.bz/gxzl>), so when we solve a problem, we tend to attach the memory of the solution to the memory of the problem. Therefore, when we encounter the problem again in the future, the solution is right there with it in our minds.

You've experienced this yourself: the first time you encountered a particular error in your code, or ran into a specific networking problem, or were faced with a server daemon that wouldn't start for some reason, you may have struggled with it. You probably hopped on a search engine and looked for other people who'd run across the same thing. But once you solved the problem, the problem and solution became encoded in your brain. The next time you ran across the same problem, you thought, "Oh, I've seen this before," and your brain served up the solution. You *learned*. But how does that physically occur inside your gray matter?

Our brains are constructed from specialized cells called *neurons*. These neurons are distributed across different sections of the brain, with each section managing something specific. There's the visual cortex, for example, which does most of the processing needed to let us see. (The material in this section on how learning happens is described in detail in *How We Learn*, by science journalist Benedict Carey [Macmillan, 2014]. Here I'll just present the big picture.)

Neurons connect to each other through *synapses* to form *synaptic networks*. Those form the basis of our memories. When we experience something, the relevant neurons "light up," representing the visual, auditory, tactile, olfactory, and taste aspects of the experience. Non-sensory knowledge also lights up neurons in the regions of the brain, such as the areas that store facts we've learned. The total collection of the synaptic network is, in essence, a *memory*. Memories can be weaker or stronger: a multi-sense experience that made a big emotional impact will often be stronger, whereas a single-sense experience that didn't make a big impact will be weaker. Contrast a memory from a special time in your life (such as a hike with a loved one on a warm autumn day in the fragrant woods, crunching leaves underfoot) with the memory of the person

you just walked past on the street, and you'll understand the difference that senses and emotions can make.

Recalling a memory also makes it stronger: "This is apparently useful," your brain thinks, "and so I'll keep it handy." A song you sing or hear a lot will be more "memorable" than one you've only heard once.

All of these built-in brain features started as survival mechanisms. Memories have a kind of "survival of the fittest" action: the memories you use the most, which result in your continued survival, are the memories that pop up the most readily when you need them. So, for instance, the memories in the forefront of the minds of early humans were things like which plants were poisonous, and where the water sources were located.

In fact, *survival* is a good word to remember. Our brains, and the way they work, evolved to keep us alive in the wild, back in the caveman days. The way our brains work supports *survival*. Today, we might not need our brains to keep us alive in the wild as often, but our cognitive mechanisms still work the same way.

That's why memories we recall and use often are stronger and easier to recall each time: your brain is built so that day-to-day information can be recalled almost instantly. Information that doesn't get used is filed away and can eventually be forgotten, because it obviously isn't needed during the day-to-day struggle to survive. Memories that resulted from an impactful, multi-sensory experience—like being chased by a predator, or experiencing a major server crash on your watch—will immediately be stronger, because the experience represented a survival event. Less-impactful memories, like whether or not it was sunny a week ago, don't contribute to survival, and so they aren't as strong.

Understanding how the brain works lets us leverage those built-in mechanisms to create more effective learning experiences. No amount of logic or reasoning will change the way our brains learn: you have to work *with* the brain to create the best learning experiences.

18.4 The value of repetition

Experiences aren't the only way we can learn, as any child who learned their "times tables" in school can remind you. If you just repeat something often enough, your brain's survival mechanism will kick in and make that memory accessible. Even today, I know that 9 times 4 is 36 mainly because my brain's survival mechanism was tricked, through endless repetition, into believing that particular memory was important.

But repetition can be used to strengthen experiential memories as well. Musicians rely on what they call "muscle memory" to play instruments. Their muscles don't actually have memories, of course; it's just the constant repetition during practice that makes those memories come up instantly. Their brains learn that "When I see this symbol on the page, I do this with the fingers."

Pounding facts into someone's head via repetition, especially for adults, can be a horrible way to learn, because many people hate repetition. It's why I personally can't

do more than plunk out a basic tune on a piano: I don't have the patience to sit through the endless repetition that learning to play a piano would require. But that doesn't mean repetition doesn't have its place. Many people do have the patience to practice scales—otherwise we wouldn't have professional musicians. But even if you—or the person you're teaching—are not fond of repetition, eventually you'll come to the point where you want them to try doing it on their own (and even someone with the patience to practice scales on the piano eventually wants to play a real song!). Some time later, make them do it again. And some time later, again. It doesn't need to be an every-single-day event, but forcing the brain to recall, filling in the bits it may have missed, and doing that every so often, will reinforce the memory and keep it sharp. After a time, there'll be no way to make the memory fade anymore . . . as anyone who's heard the "It's a Small World" song can attest!

18.5 **Getting in and doing it**

Because our memories can encompass abstract knowledge and facts along with sensory impressions, the memories with the most "factors" tend to be stronger. Hearing someone tell you how to change the oil in a car is one thing. Watching a video of the process would be even better, because the visual cortex of the brain can get engaged, and because there'd likely be an audio element. But actually doing it on a real car involves touch, sight, sound, facts, smells, and if you're unlucky, taste. That's a combination of senses that the brain won't soon forget.

That's one of the reasons apprenticeships have always been, in my mind, a more powerful way of learning than the "book smarts" most higher education programs focus on. In an apprenticeship, the apprentice is right there in the thick of it with the master, and likely with skilled journeymen. They're creating powerful memories using not only abstract facts, but all five of their senses. Reading about being a blacksmith is a very different experience than actually pounding a piece of white-hot metal with a hammer.

When I was an aircraft mechanic apprentice, we did have classroom time. About once a quarter, we'd spend a week, or two at the most, learning the theory of the aircraft's design and operation. But then we went back to the shop floor and worked on those very same components. Nearly thirty years later, I can still cite some of the abstract, theory-of-operation facts I learned, because they're inextricably connected to some very vivid, sensory-based, hands-on memories. You only have to tear down an F-14 hydraulic mixer valve once before you have some very vivid memories, I assure you.

So when you're teaching someone, get them involved as early, and as often, as possible. It's better to briefly explain what they're about to do, and then set them on it (supervising as necessary), than to try and explain all the background theory, all the tasks, and everything else up-front. Just get in it. For example: when you're helping a colleague with a problem, rather than shouting "MOVE OVER," sliding into their seat, and fixing the problem yourself, make them do it. It'll take longer to solve the problem, but they'll learn the solution. Watching you do it involves one sense; doing it themselves involves multiple senses and will form a more durable memory.

18.6 Why analogies work . . . and how they can fail

Analogies are some of the most powerful ways we have to teach. In teaching, we often use an analogy to take something the learner is already familiar with, and use it to explain something they don't yet know.

For example, you're probably familiar with cars. But perhaps you don't know about *object-oriented programming*, or OOP. Basically, OOP treats everything in the computer as an object, and an object is a lot like a car. Cars have properties, right? The make, model, color, engine size, and so on, are all *properties* of the car. Software objects have properties too, like the version number, the manufacturer name, and so on. As a programmer, you can examine those properties to learn about the object, and even change some of those properties to modify what the object does. Imagine being able to change the color of your car by just changing the "color" property—that's what you can do in software!

I've just used an analogy to a common object to briefly explain a computer-related concept. Similarly, one of the biggest values that you, as an individual person, can bring to the people you teach is a set of analogies that work for them. Remember, we all come from a unique background, maybe even different cultures, and we all have different past experiences; creating analogies that "speak" to a particular student usually requires that we share, or at least are aware of, that student's same background, culture, and set of experiences. This is why I can't be a great teacher to everyone; I lack the diversity needed to construct analogies for everyone. What if you come from a culture where cars weren't common? Most of my analogies would fail. Teaching, in many respects, is nothing more than taking knowledge that someone else has created, which you have subsequently learned, and then "repackaging" that information into analogies that your particular audience will understand.

Be aware that all analogies, even (and perhaps most especially) the best ones, eventually fall apart. My software-objects-are-like-cars analogy will get you to a point in your understanding, but at some point the analogy quits working. Analogies often require us to oversimplify some aspects of what we're teaching, or to temporarily ignore certain details. That's fine. People can't learn All The Things all at once. So we can use analogies to get them past a certain point, and then either switch analogies, or drop the analogies completely, to continue. We can go back and revisit things: properties of software objects aren't exactly like the properties of a car. In software, you can have some properties that are collections, which means they can contain other objects. It's a bit like if your car had a "tires" property, which contained a collection of Tire objects that each represented a tire on the car. That's a fine approach, and it's important to keep in mind that analogies are meant to serve a purpose, and then be set aside.

18.7 Do it like Socrates

In the mid-1960s, instructional designer Jerome Bruner described an instructional design technique called *constructivism*, which today is sometimes called *constructionism*. It puts the teacher in the role of a facilitator: they're not there to impart facts or

information, but instead to ask the student questions and direct them to identified resources. It's an extremely effective way to teach and learn. For example, public schools in the United States struggle to teach software development because it's so difficult to get qualified teachers to do it (you can make a lot more money actually developing software, or teaching it in the technology training industry). So in 2000, I developed a high school textbook on software development that was designed to be facilitated, rather than taught. The classroom teacher would introduce the student to documentation and other resources, show them an example provided in the book, and then ask them to perform a task. When the students invariably got stuck—the book was designed to lead them into getting stuck—the teacher would basically just point to the documentation and other resources, and start asking the student a series of provided questions.

This is a slower way to learn, to be sure. The Socratic method of asking someone questions, rather than answering questions, invariably takes more time. But it requires students to assemble their own mental models for what they're learning, rather than just ingesting facts that are handed to them. Students' brains create their own equivalencies and understandings, and the resulting synaptic networks are much stronger. More importantly, students become more effective self learners, which in the technology industry is possibly the most important job skill they can have.

Adult students can get frustrated by the Socratic method. They're often facing a problem that they need to solve, and there's often an element of time pressure on them to get it done. Standing around asking them questions, instead of just giving them the answer, can feel punitive or mocking. But you can and should use this teaching technique whenever you can. Be up front about what you're doing: "I know you're in a rush, but this is important, and it's going to be more effective if you construct the solution yourself. I'm going to ask you some questions to get you thinking in the right direction, and I'm right here with you. This will be worth the extra time."

And then you start asking questions. To do so, think of what you know the answer to be, and then ask questions to get your "student" there. For example:

THEM: "The server isn't responding."

YOU: "How are you trying to reach it?"

THEM: "With the web browser, but it just says server not reachable."

YOU: "Are there other ways you could try to reach it?"

THEM: "I tried a ping, but it said it couldn't resolve the name."

YOU: "How does name resolution work?"

THEM: “It uses DNS.”

YOU: “Are you sure DNS is working?”

That kind of discussion definitely takes longer than you simply telling them, “Yeah, the DNS server is down and that’s why nothing is working,” but going through the question-and-answer process lets them construct their own mental model of how all those moving network pieces fit together. You’re not just providing solutions, you’re teaching them to construct their own solutions, and that’s a far more valuable outcome in the long term.

18.8 **The importance of sequencing**

As you teach, it’s really important to *sequence* what you teach. By sequence, I mean organizing and ordering the way you present the information.

Okay, we’re going to cook a meal. This is going to be a basic steak-and-potatoes meal, so we’ll only have three or four main ingredients. You do need to be careful, because once we start, the stove is going to be hot, as will the pans we use. And at the end, you need to make sure you give the steak a few minutes to rest before you serve it, because it will actually still be cooking a bit when we take it off the heat. But first, we need to assemble our ingredients, and that’s going to require you to learn to julienne a vegetable! But let me start by explaining the history of cattle husbandry in the United States.

This so-called cooking lesson is so mis-sequenced that anyone listening to it would be completely justified in throwing up their hands and going to a restaurant to eat. When organizing your material, consider these three guidelines:

- *Don’t cover abstract concepts unless they directly relate to something practical that you are just about to teach or have just taught.* The history of cattle in the United States is not directly relevant to cooking a steak, so don’t get into it now.
- *Cover material in the order a student would encounter it.* In this example, start by setting up the ingredients. Teach them how to julienne as part of the prep process. Continue from there, eventually arriving at cooking and resting the steak.
- *Do not bring up cautions or warnings until just before they’re relevant.* Otherwise, you’re asking someone to remember an abstract fact, disconnected from any practical use, until later, when it suddenly becomes mission-critical. Human brains aren’t good at that. So warn them about the hot stove when they turn on the heat, not when they are gathering the ingredients.

Also, don’t forget that you cannot prevent failure. Not ever. You can’t take an approach of, “I’m going to start by telling you all the things I wish I’d known before I made my

first steak,” because you’re just stacking abstract facts rather than getting into the task. And failure often creates a memory, so it can be a useful learning experience.

Hey, see there where your pan is smoking? That’s because you used the wrong kind of oil. Each oil has a different smoke point and flash point, and you need to use one that’s suitable for the heat you’re cooking with. Let’s set that pan aside to cool, and start over.

That’s a perfect example of controlled failure. Rather than make a big deal out of oil flash points and smoke points up front, you just let nature safely take its course. Once the problem occurs, you offer a solution. Human brains love problem-solution; they do not love solution-no-problem as much. We learn from mistakes, so you need to sequence in the right mistakes so that the learner has an easier time ingesting all the info.

18.9 **Rest time is crucial**

Finally, remember that the human brain can physically only digest so much new information in a given period of time. If we’re *doing* something with the information, as in getting hands-on with it right away, we can learn better, and learn more, in the course of a day. When we read or listen to abstract facts not connected to an activity or sensory experience, our ability to intake information is really limited. So rest time is crucial.

Sleep, the ultimate rest, is also crucial to learning. When we sleep, our brains organize our memories. Our brains decide which synapses get stronger, and which ones get marginalized, based on how we’ve been using those memories. New memories get connected to relevant older memories. We *need* time for this to occur.

That’s yet another reason why I prefer apprentice-style learning to a week of all-day in-classroom learning. Apprentice-style learning builds better memories, and it spreads the learning out over a longer period of time, giving my brain more time to cope with the input. Classrooms can be a firehose of information with little connection to my real world, so I tend to forget a lot of what I learn.

Being deliberate with rest time

When I designed the *Month of Lunches* book series for Manning, I employed as much cognitive science as I could. Each chapter of a *Month of Lunches* book is designed so that the average adult can read it in about 45 minutes—a time period I came to by researching the average adult reading speed in words per minute. Even though the chapters are short, you’re meant to read one chapter a day, because each chapter focuses on a single topic and then relies on your brain’s “rest time” to synthesize that topic with the previous ones and set you up for the next one. Readers who blow through the whole book in three days have told me they didn’t retain as much, and I’m never surprised. (The chapters also include frequent exercises and labs, intended to get readers to do something hands-on with the new concepts they are learning—another way to get the lessons to stay with them.)

18.10 Further reading

- *Instructional Design for Mortals*, Don Jones (independently published, 2018)
- *How We Learn: The Surprising Truth About When, Where, and Why It Happens*, Benedict Carey (Macmillan, 2014)

18.11 Action items

For this chapter,

- Make a list of all the things you feel you could teach others. Don't limit that list just to your technology field, either! Maybe you can teach cooking, or how to change the tire on a car, or any number of things. They're all valuable to someone who doesn't know them. The idea here is to start convincing yourself that you have plenty to offer, provided you find the people who need what you know.
- Create a plan for teaching. Pick a topic, define an audience who needs that topic, and decide how you'll engage that audience. Perhaps you'll write a series of blog posts explaining how Python works to an audience of PowerShell users. Maybe you'll teach some of your co-workers what you've learned about how your employer's business operates. Maybe you'll teach the kids at a local youth club how to cook a basic meal. Whatever the topic and whomever the audience, just start teaching.

19

Be prepared for anything

If the COVID-19 crisis taught us anything, it's that *anything* can happen to your job, and in some situations even a well-managed career isn't enough to completely save you. That's why it's important to plan for those situations, and have some idea of what to do when they arise.

19.1 *What can happen?*

To begin preparing for “anything,” which might include suddenly finding yourself without a job, you first need to contemplate what kinds of problems might occur. That way, you know what you’re attempting to mitigate. This list will vary tremendously from person to person, and especially from region to region, but you should consider events like these:

- Getting laid off, for any number of reasons outside your own control—like an economic crash that forces your company to reduce headcount.
- Having an accident and becoming injured or extremely ill, forcing you to leave your job or take significant unpaid time off work.
- Running out of work, for independent contractors.

Spend some time thinking about what might go wrong, because these are the things for which we’ll need a back-up plan. But sometimes, “preparing for anything” can mean preparing for something *good*, like an amazing job opportunity that just comes out of nowhere! Since that type of event doesn’t usually result in panic, a loss of income, and other negative elements, it’s easier to plan for—but it’s still worth planning for it.

19.2 Basic preparedness goals

The idea behind preparedness is to imagine *reasonable* situations that might occur, and then make plans to mitigate the negative fallouts. For the purposes of this book, I'm limiting the discussion to situations where you've lost a job, gotten injured, or have simply run out of work as an independent contractor. Those situations all have some common elements, the primary one being *loss of income*.

With some reasonable situations in mind, and with an understanding of their primary negative impacts, you can start to develop preparedness goals. For example, with any situation that leads to loss of work or income, you may need plans for how to

- Cut back expenses and protect cash reserves
- Quickly obtain new employment or work

You may imagine different situations for yourself. For example, given recent history you might imagine a situation where a global pandemic breaks out, basic supplies like fresh food become limited or difficult to access, and you wind up having to work from home. In that situation, you might adopt different preparedness goals, such as

- Ensuring the family can remain safely at home as much as possible
- Ensuring sufficient non-perishable food is on-hand to mitigate any supply shortages
- Ensuring a space can be converted to a work-from-home office

You'll have to decide—based on your own experiences and concerns—what's important to you, what to prepare for, and what preparedness goals to set. For the remainder of this chapter, I'll focus on the job- or work-loss situation, and offer some suggestions for how to think about preparing.

19.3 Cash on hand and credit

The first element of my own backup plan is *cash on-hand*. My family has a well-defined budget in the event that I am unable to work or am out of a job. That budget eliminates every possible discretionary expense (like Netflix or other subscriptions), and cuts back on required expenses as much as possible. We might reduce our internet speeds to a cheaper plan, for example, and eliminate all dining out in favor of making less-expensive meals at home. We strive to keep six months' worth of that budget readily available for immediate use if needed. “Readily available for immediate use” can mean a variety of different things, some more immediate than others:

- Cash in a savings account
- Access to a secured line of credit, such as a home equity line of credit
- Penalty-free access to funds in a retirement account

I don't regard my stock market investments (mainly my retirement account) as “readily available.” While I could certainly sell stocks to generate cash, if my disaster was caused by an economic crash then my ability to sell those stocks might also be negatively impacted. The point of this list is to know where you can *absolutely* get enough cash, *immediately*, almost no matter what has happened.

And I want to acknowledge that not everyone has access to a supply of cash-on-hand for emergencies. I certainly didn't when I was starting out, but it was one of the first things I prioritized in life, even above furniture and video games. Gradually building that supply (and it took a few years) gave me the flexibility I needed in the workplace, and the peace of mind I wanted.

Also be sure to consider how your family could access this cash if *you* were unable to do so, perhaps through a serious injury. If you're relying on funds from a retirement account, for example, make sure your family is able to access those funds (speak with an attorney or banker to understand how to set that up).

"Six months' cash on hand" is a number I came up with that makes *me* and my family comfortable, and it's a number that our financial adviser tells us is a common target. For freelancers, some advisers recommend nine or even twelve months. But we're reasonably confident that I could find some kind of work in the six-month time frame, and we use the other elements of our backup plan—which we'll cover in the upcoming sections of this chapter—to supplement that number. The standard guidelines suggest having access to cash to live on for three to six months, but you'll need to come up with your own number that makes sense for you, your family, and your own level of confidence in your ability to find new work.

Also consider your available credit. In the event of a disaster, you may want to conserve your cash on-hand as much as possible, so that you can stretch it and make it last longer. Credit—in the form of credit cards, for many of us—can be one way of making your cash last longer. But to use that credit means you have to ensure it's *available*. My family's policy, based on our financial adviser's recommendation, is to have at least 30% of our credit lines free at all times—we never max out our credit cards unless it's a dire emergency, and then we prioritize paying them back down under the 60% mark. That 30% number is a common rule of thumb, and one we feel comfortable with and capable of managing. Your circumstances and comfort level may differ, and consulting with a financial adviser can help you settle on your own number. In a true disaster, we'd feel fine maxing out our credit lines *and* making only the minimum payments on them, so as to better conserve our cash reserves. Let me stress that this is not the usual approach, and many would advise against it. Depending on the terms and interest rate of your credit cards, this could get very expensive over time. You'll need to review your credit terms to decide if that's an acceptable approach for you. My point is to have a solid plan, well in advance, of what you'd do if you *had* to.

19.4 Social safety nets

Social safety nets refer to government programs designed to protect its citizens in times of need. These safety nets often include various ways of providing money, medical care, or other resources to people affected by natural disasters, economic downturns, personal job loss, or other emergencies.

As part of your preparedness plan, make sure you understand whatever social safety nets you may have available to you. For example, in the United States, full-time

employees who are dismissed without fault (that is, laid off or furloughed) can usually collect from a state unemployment fund. Know how this works ahead of time, including where to register, how much the benefit covers, and the length of any waiting periods. You should have an “emergency plan” that includes the websites and forms you will need to use to file a claim. In the United States, “how this works” differs from state to state, and of course other countries have completely different systems. Start with your government’s websites as you begin creating your plan.

Remember: the time to discuss your airplane life vest is *before* you need it, not as the plane is going down. The time to understand your social safety nets, and *how to use them*, is *before* you need to use them, not the day after you get laid off.

And just as with your cash on-hand plan, make sure your family knows how to file and access your social benefits on your behalf, if needed. Having a written, documented plan, stored somewhere that everyone in the family can access it, is a great way to make a stressful situation a little more bearable if the time comes.

19.5 Insurance

Most adults are familiar with the idea of life insurance, and it can be a valuable tool to have. My own financial adviser recommended I take out a term life policy, with a term long enough to carry me through to retirement age. His theory is that by retirement age, my actual retirement plan will be ready to go and so I won’t need the insurance any more. If you buy term life insurance when you’re young, it’s amazingly cheap (a healthy 25 year-old in most parts of the United States can buy a \$2 million policy for a couple of hundred bucks a year). Term life costs a fixed amount each month, and pays out a fixed benefit until the term expires. Once the term expires, you stop paying premiums and the policy no longer pays any benefit.

GET GOOD ADVICE! I’m not a financial adviser and don’t play one on TV: I’m offering you these suggestions as a *starting point* for a conversation with a qualified adviser. They’re the ones who can look at your specific situation and make specific, actionable suggestions. If you want to use my example as a starting point, it can help them understand what you’re looking to solve for.

But death isn’t the only disaster that can befall us, and there are a variety of different insurances that you can consider for your backup plan. Before we go into those however, I want to caution you against relying *solely* on insurance provided through your employer—that goes for any insurance, including life and disability. Employer-provided insurance can be a wonderful *supplement*, and in the United States it’s often inexpensive. But if you lose your job, it goes away, and losing your job is one of the disasters we’re trying to mitigate. Try not to rely on employer-provided insurance as a primary part of your plan.

With a reminder that I can only speak for the situation in the United States, but that the basic concepts are reasonably global, here’s some of what you and your adviser can consider in the insurance world:

- *Life insurance* comes in two broad flavors, whole life and term life. Whole life tends to cost more, but it stays in effect until you either use the policy or cancel it. If you cancel it, it'll often have a cash value that can be returned to you. Term life is valid for a specific period of time, and is often much less expensive.
- *Mortgage insurance* is a little bit like a term life policy that extends for the duration of your home mortgage: you pay a monthly percentage of your mortgage, usually X to Y percentage, and the insurer agrees to pay off your mortgage for you. That way, if something happens to you, your home is paid off and it's one major thing your family doesn't have to worry about.
- *Disability insurance* is designed to replace some or all of your income if you are partially or fully disabled and unable to work in your accustomed field. Short-term disability is one thing I don't mind buying from my employer, as it's only designed to cover a few months' of disability and kind of implies I'd be on the job to begin with. Long-term disability policies usually kick in *after* a few months have passed, and are usually structured to pay out for life, if you're permanently disabled.
- *Medical insurance* helps offset the horrific cost of medical care. In the United States, individuals or employers currently purchase it on the open market through private health insurance companies. It's usually provided to full-time employees by your employer, although in the event you lose your job you should have a plan for obtaining replacement coverage, which varies widely depending on the state where you live. Do that research *now*, while you're making your backup plan. (In some other countries, health insurance costs may be partially or fully funded by the government.)

According to numerous studies (<http://mng.bz/5WRz>), medical bills are the biggest cause of bankruptcies in the United States. Whether or not you have health insurance, it's easy to fall into debt problems through a medical emergency, where you not only start racking up huge medical bills, but may also lose your ability to work. To help avoid debt through medical emergencies, my financial adviser has recommended a long-term disability policy, which I carry to this day, and relying on my cash on-hand fund to cover the short-term disability period. In other words, use the cash until the long-term policy kicks in. I also know how to sign up for independent health insurance if needed (in my home state of Nevada, the state runs a website that offers policies), and I have a term life policy to carry my family through until my retirement age. That term life policy is also sufficient to cover our mortgage, so we don't carry separate mortgage insurance.

19.6 *Prestaged job hunt tools*

As with all disaster planning, the time to get ready to job hunt is not when you need to job hunt, but *before* you need to job hunt. Here are the top tasks to stay prepared for conducting a job hunt:

- Keep your LinkedIn profile up to date and ready to help you start creating résumés.
- Watch your personal brand and social media footprint, so that any publicly visible portion of your life is professional-looking and “job-hunt ready.”
- Periodically browse jobs in your area (or across the region or country, if work from home is a common option in our trade) and your field to see what employers are hiring for these days. This lets you keep your skills relevant, so that if you *have* to job hunt, you’ve got at least some of the right skills for the current marketplace.
- Continually grow and maintain your professional network so that it’s there to help you if you need it.
- Make sure you’ve got a set of clothes that fit and are appropriate for any kind of job interview. For me, this is a dress shirt, dress slacks, blazer, and tie.

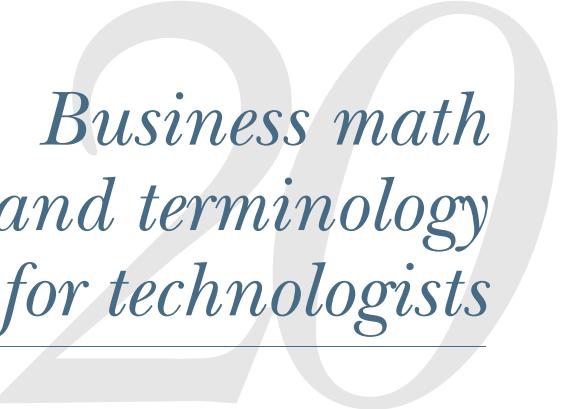
These preparedness activities are different from the ones discussed elsewhere in this chapter. Things like buying insurance policies or documenting unemployment benefits are a one-time task you can do, file away, and not worry about till you need them. But staying prepared for the job hunt is an *ongoing* activity. You should view it not *just* as disaster preparedness, but as a normal, daily part of any healthy career. And here’s another bonus for keeping your job tools ready to go: if an opportunity (not an emergency) pops up, you’ll be primed and ready to seize it!

19.7 Action items

For this chapter, I’d like you to start considering *what could go wrong*, and building a plan to prepare for and mitigate it:

- Make an appointment with a financial adviser. They’ll have experience with other people who’ve gone through tough times, and they can provide good advice that’s specific to you, where you live, and your life situation. Explain the types of emergencies you’re hoping to plan for, and listen to what they say.
- Note that, in the United States, I prefer a fee-only adviser, which is one who collects a flat fee for their time. The alternative is a fee-based or commission adviser, who makes money from the products they sell you.
- Many countries have certifications for financial advisers, and an internet search should turn up appropriate local agencies no matter what country you live in.
- Look closely at your monthly expenditures, and develop a budget for emergencies. That budget should clearly indicate what gets cut and what doesn’t, so that in an actual emergency you can quickly start to take action. My plans even include the websites and phone numbers of services and subscriptions I’d want to quickly cancel or scale back.
- After speaking with an adviser, start looking at the insurances you might want to purchase to help cover emergencies. Remember that dying isn’t the worst, or even necessarily the most common, thing that can financially affect your family.

- Research the social safety nets available to you, and begin documenting what they are, what they offer, how you apply, and how long it usually takes them to begin. Keep all of this documentation in a place where your family can quickly access it at need.
- Implement a monthly or quarterly “job tools update” program, so that you can keep your job-hunt tools in ready-to-use condition. Remember, sometimes these tools will be used for a good reason, like taking advantage of an unexpected opportunity!



Business math and terminology for technologists

We're technically leaving the world of soft skills now, but moving on to skills and knowledge that will still be of benefit to any technologist looking to create a more successful career. The world of business—and that includes nonprofit businesses and many government organizations—has its own language and its own mathematics. If you're going to play the business game for your livelihood, it can be an immense help to understand a bit about how businesses talk about themselves, how they measure themselves, and how they handle their internal math.

I'll note that this chapter does gloss over many of the fine details of the topics I'm covering in an attempt to better convey the high-level importance of these terms and concepts. Consider this to be just a starting point, and you should definitely think about exploring these topics in more detail as you flesh out the skills you need to succeed in your career.

20.1 How much do you cost?

After many years in the working world, I have come to the conclusion that the most valuable piece of information I can have as an employee is knowing how much I cost the company. (When possible, I like to know how much everyone else costs too, and not just because I have salary jealousy—kidding!)

Here's why: let's say I spend a hundred hours, over the course of a couple of months, working on a project that will automate some important business process. I might wonder, was I right to spend that time? Did I spend too much? Could I have spent more? Should the company be thankful? *How* thankful? Knowing what I cost the company can help me answer these questions.

Let's say I make a salary of \$100,000 (I like easy numbers for this explanation). My actual cost to the company is much higher: in the United States, the company has to pay payroll taxes (those are on top of the income taxes withheld from my check that I pay) too. They also cover my healthcare, contribute to my 401(k) retirement plan, and cover the cost of stock options and other benefits. My existence as an employee, along with my fellow employees, also requires the company to employ Human Resources staff and pay for other overhead costs,. To account for all this, most companies in the United States add about 40% to the base salary, which means I'm actually costing about \$140,000 a year as a "fully loaded" salary. *Fully loaded* refers to the base salary I see on my paystub, plus all the employer-paid taxes, benefits, and similar costs.

There are roughly 2,000 working hours in the year. Again, that's a United States-specific number. Here's how I figure it:

- There are 52 weeks in the year.
- Most employees have two weeks paid vacation, so that leaves 50 weeks.
- Most US companies observe about five paid holidays like Christmas, so that leaves 49 weeks.
- There are about 40 official hours per work week: $40 \text{ hours} \times 49 \text{ weeks} = 1,960 \text{ hours}$, which we can round up to 2,000.

So dividing my fully-loaded salary of \$140,000 by around 2,000 hours, I get an hourly fully loaded rate of \$70. That's what I cost the company per hour.

Now, going back to my automation project, at \$70 per hour, this means my 100-hour time investment cost the company about \$7,000, plus the cost of however much coffee I drank during that time, the cost of the electricity I used, and other overhead that's too complex to go into. So was the investment of my time worth it?

That all depends on the *return* the investment creates. Let's say the business process I automated had formerly been done by an entry-level help desk person, who spent about 10 hours a week doing that task. They make a base salary of \$50,000, or a fully-loaded hourly rate of \$35. Doing the task manually therefore costs \$350 a week, which means we'll recoup my \$7,000 investment in 20 weeks—about five months. That seems like a pretty solid return on investment: after 12 total months, the company will have saved \$11,200 in time.

PUT THAT IN YOUR RÉSUMÉ When you create savings like that, track it. That's solid résumé material: "Saved previous employer \$20,200 annually through a \$7,000 one-time investment in automation."

I find it incredibly helpful to do this kind of calculation. First, it helps me make smarter business decisions, such as where should I spend my time and what projects make fiscal sense for the company. It helps me better communicate the value I bring to the company, in terms that the company itself understands and can relate to.

WHAT ABOUT WHAT OTHER PEOPLE COST? Many companies don't publicly share peoples' salaries, so it can be difficult to calculate costs across a team. In those

cases, you can sometimes work with your Human Resources and/or Finance departments to get a “model salary.” That won’t represent what any one person gets paid, but rather an average or other “generic” salary that you can use for calculations.

20.2 **Reading a P&L statement**

A Profit & Loss Statement, or P&L, is one of the high-level tools that you can use to understand a business’ finances—and often, its most basic motives. Now, I want to once again point out that I’m going to be really glossing over some fine-print details in my explanation of these things. This isn’t intended to be a CPA-quality explanation; it’s meant as an orientation.

A P&L can tell you a *lot* about a business. So much that most businesses don’t want them floating around in public (and publicly traded companies often aren’t allowed to share them broadly outside of certain legally specified engagements). Even a high-level P&L will show you where most of the company’s revenue comes from, and where most of it is spent (payroll is often an enormous percentage of the overall expenses). A detailed P&L can reveal much more, and help you truly understand how a company runs.

It’s worth asking if your company is allowed to share a high-level P&L with employees. If it is, consider showing an interest and reviewing it. You’ll often run across stuff that doesn’t seem to make sense, and that’s where you can truly dig into the details that a company deals with. You can see an example P&L statement for a fictitious company in figure 20.1.

Every business does these a little differently, but broadly speaking, the top section will usually include what I call the Good News. That is, *revenue*.

20.2.1 **Revenue**

At the top of every P&L statement is the company’s *income*, which includes *revenue*. Other types of income can include interest earned from bank savings accounts, but revenue is the big deal because it’s the money you make from actually running the business.

You first want to understand if the company operates on a *cash basis* or *accrual basis*. Sometimes the P&L will state that (my example doesn’t), while at other times you’ll need to ask someone in company leadership. Most small companies use a *cash basis*, which means that the cash they have in-hand is real cash and counts toward revenue, and the expenses they’ve paid are actual settled pay-outs of cash, and count as expenses. This is how every private household operates, and it’s the easiest to understand.

Accrual basis is a little more complicated, and it’s what most big companies use because it gives a more well-rounded picture of a company’s financial health. Something counts as revenue when you’ve *invoiced* for it, or *billed* for it, even if you haven’t gotten the cash yet. Something’s an expense when you receive an invoice or bill for it, even if you’ve not yet paid it. Basically, cash basis means you count the actual cash and

Rock Castle Construction Profit and loss December 1 – 15, 2015		
	Dec 1 – 15,15	% of income
Ordinary income/expense		
Income		
40100 · Construction income	57,238.91	99.5%
40500 · Reimbursement income ►	285.00	0.5%
Total income	<u>57,523.91</u>	<u>100.0%</u>
Cost of goods sold		
50100 · Cost of goods sold	4,563.81	7.9%
54000 · Job expenses	21,477.46	37.3%
Total COGS	<u>26,041.27</u>	<u>45.3%</u>
Gross profit	31,482.64	54.7%
Expense		
60100 · Automobile	81.62	0.1%
62100 · Insurance	1,214.31	2.1%
62400 · Interest expense	32.58	0.1%
62700 · Payroll expenses	15,117.86	26.3%
63100 · Postage	69.20	0.1%
63600 · Professional fees	250.00	0.4%
64200 · Repairs	175.00	0.3%
64800 · Tools and machinery	810.00	1.4%
65100 · Utilities	122.68	0.2%
Total expense	<u>17,873.25</u>	<u>31.1%</u>
Net ordinary income	13,609.39	23.7%

Figure 20.1 An example P&L statement showing revenue, costs, and expenses

expenses that have occurred in a given time period, while accrual counts actual plus expected income and expenses.

So who cares about cash versus accrual? It can make a big difference on things like taxes. When a cash-basis company receives an invoice from a vendor, it doesn't "count" until the invoice is actually paid. If you receive an invoice at the end of one tax year, but don't pay it until the next one, then you can't deduct the expense from the prior year—it's deducted in the year in which the expense is actually paid. That's why a lot of cash-basis companies rush to pay invoices at the end of the year. Accrual-basis companies are the opposite: they often rush to *get* invoices at the end of the year so they can deduct the expense, but don't *pay* them until the next year—which can sometimes frustrate the people who sent those invoices!

There are different types of revenue, too. The most common one is *ordinary income*. This is, quite simply, the revenue you make from most forms of business, where you sell a product or perform a service and then get paid for it.

PERCENTAGES VS. ACTUAL AMOUNTS In figure 20.1, you'll notice that I have a big arrow pointing down over the "% of income" column. For many businesses, these percentages matter more than the actual monetary values. For example, payroll in this example is 26.3% of income, making it the single largest expense. If you needed to reduce expenses, that big chunk might be a first place to look.

Subscription income is a bit different. Let's say you run a service that charges \$10 a month, but you actually bill for it a year at a time. So you've sent a \$120 bill to a customer, and they've paid it. You can't actually "recognize" all \$120 of that as revenue as soon as you get it; instead, you have to recognize \$10 per month, because that's what you're actually earning. This is all due to a set of rules called the Generally Acceptable Accounting Practices, or GAAP.

Cost of Goods Sold, or COGS, is another form of revenue, and often shows in the Good News, or revenue, section of the P&L. COGS itself is a bit complicated to explain. Let's say you buy a few thousand small, empty cans, and a few tons of raw nuts. Your company then fills the cans with raw nuts and sells the completed package. Both the cost of the nuts and the cost of the cans are COGS, or "the cost you need to incur to sell the goods that you sell."

From a tax perspective, you can't write off those costs until you've completed the accompanying sale. So if you sell half the nuts in the year you bought them, then you could write off half the cost of nuts and cans in that same year, and you pay taxes on the rest. If you sold the rest the next year, then you couldn't claim those expenses until that year. This is why some companies hate inventory so much: you're basically paying income tax on the money you used to buy the inventory until you actually sell it. It's why car dealers have end-of-year blowout sales—so they can sell the inventory and write off the costs.

Many companies have other kinds of income, such as interest on bank accounts or investments, recovered debts, and the like. It should all show on the P&L in the revenue section.

All of a company's revenue, minus the cost of whatever it sold, results in its gross profit, and this total is also listed on a P&L. So if a business spent \$1 million on nuts and cans, sold them all in a year, and generated \$2 million in revenue, it would have \$1 million in gross profit, and a 50% *gross profit margin*, or gross margin. Most industries have more-or-less standard gross margin goals, and it's one way to compare your basic profitability with your industry peers.

20.2.2 Expenses

Next is the Bad News section, or Expenses. This is money a company spent *aside from* the cost of the goods you sold. This includes advertising, payroll, taxes, office supplies, and pretty much every other penny it spent. This also usually includes discounts they've offered to customers, which is why so many CEOs hate discounts: they're like

this vast negative number on the “bottom line,” right before you find out if you’re bankrupt or not.

Gross profit, minus total expenses, is your net profit or net loss, which is where a P&L gets its name. Again, most industries have benchmarks for their net profit margin (expressed as a percentage), and most companies try to meet or beat their industry norm.

Take Apple. In 2007, they introduced both the first iPhone and the first iPod Touch. Both devices ran iOS v1.0, but when Apple released the first iOS update, iPhone users got it for free, while iPod Touch users had to pay \$20 or so. How come?

At the time, Apple was getting monthly payments from AT&T, and so there was a small amount of monthly revenue attributed to each iPhone. Apple could say that the iPhone was complete as initially shipped, and that the monthly incremental revenue was what paid for the subsequent “improvements” in the form of a new iOS.

But they couldn’t say that with the iPod Touch because they’d charged full price for it up front. Releasing an “improvement” meant that they’d *increased the value of the product*, which meant having to go back and restate all their revenue. Much like on a subscription service, they couldn’t claim that the original \$250 or whatever was all revenue when the product wasn’t “complete” at the initial release; some of the “new value” from a new version of iOS would have to correspond to revenue. To simplify the bookkeeping, they just charged for the update so that the update was “paid for” by its own revenue.

It was a few years before Apple revised the way they handle bookkeeping and how they attribute revenue so that they didn’t have to do that anymore. A glance at their earlier P&Ls would have revealed the oddity and driven questions that revealed all those interesting behind-the-scenes details.

20.3 Averages

Averages are useful things in lots of pursuits, and business is certainly one of them. But people often misunderstand averages. I once read an article about men’s razor blades. The reporter asked a representative of Gillette how long, on average, a blade lasts. The person’s reply was along the lines of, “Well, everyone is so different that an average isn’t meaningful.” Which will come as a huge surprise to every statistician ever because that’s the whole point of averages.

There are actually three kinds of averages. Each of these seeks to take a set of numbers and come up with a “middle ground” that represents the entire group. A *mean* does that by adding up all the numbers and dividing by the quantity of those numbers. Also called the *arithmetic average*, it’s the one most people are thinking of when they say “average.” Its downside is that it can be artificially dragged one way or another by outlier values. You might have 99 values at 50, for example, and 1 at 7,000,000. That’s a mean of over 70,000, but 70,000 is in no way representative of a group of numbers that are mostly at 50. So in looking at averages, you really need to look at the underlying data to understand how many of the values are actually clustered around the mean.

The *median* is the middle point of a sample, where half your values are above the middle and half are below. This is good for finding a literal “middle ground.” It’s actually a bit better of a value than the mean for a lot of business situations because it automatically takes outliers into account. It does diminish those outliers, though, because a far-flung outlier value won’t “move” the median any more than a “closer-in” outlier.

The *mode* is simply the most common number in the sample set. In a set with 1, 2, 2, 3, 4, 5, 6, and 7, the mode is 2, because that’s the most common sample value. So when might you care about averages, from a business perspective?

- If you’re looking at developer productivity, you may look at metrics such as number of code commits where all unit tests passed. Obviously, different developers are going to have different levels of productivity, in part depending on the type of project they’re working on. A *median* can be useful for getting an idea of where the entire organization sits as a whole. You can then look “above and below the line,” that is, above and below the median, to start understanding why those numbers are what they are.
- If you’re looking at server uptime, you might look at the *mode* of the number of days a given server is up or down within a specified time period, like a month. If the most common number of “up days” in a month is 28, you could then focus on servers with *less* up time to understand why they’re different.
- If you’re looking broadly across the organization and wanted to measure the average amount of paid time off people are taking, you could look at an arithmetic average. In most organizations, paid time off won’t have a lot of outliers, and so a mean and a median might be close. And of course, you could look at both the mean and median, and if they differ greatly, you’d know that you probably have some outliers to consider.

There’s a fun book called *How to Lie with Statistics*, by Darrell Huff (W.W. Norton & Company, 2010), that I recommend. It’s a great look at how people bend numbers and psychology, and can make you a lot smarter at business math.

20.4 OpEx and CapEx

Broadly speaking, businesses incur two types of expenses: *operational expenses*, or OpEx, and *capital expenses*, or CapEx. Understanding how each of these works, and how your company approaches them, will give you some insights into some of the business decisions your company makes.

20.4.1 Understanding the two types of expenses

Operational expenses are recurring expenses that are required in order to run your business: rent, payroll, utilities, recurring services you purchase from vendors, and so on. Operational expenses do not increase the value of your company. For example, just because you double your payroll doesn’t mean you double the value of the company. By doubling your payroll, I would presume you hope to double (or more) the

output of whatever your company does, and that might increase the overall value of the company.

Capital expenses are either one-time expenses, or expenses that are paid for over a fixed number of payments. Buying new machinery (like computers!), building a building, or buying another company would all be capital expenses. These generally increase the value of your company by whatever the new capital investment is worth. That is, if you spend \$1 million building a new warehouse, then presumably your company is worth \$1 million more, because in theory you could always sell the building, converting that asset to cash.

Companies often pay for big capital expenditures by taking out a loan. In that case, the principal of the loan is a capital investment, while the servicing of the loan—that is, the interest you pay on it—is an operational expense.

Different companies feel differently about OpEx and CapEx, and their feelings are often driven by the industry they’re in. For example, a small technology startup might prefer to make as many expenses as possible OpEx: they’d rent office space rather than build it, and they’d deploy their services to cloud providers rather than building a data center and buying their own servers. That approach might cost them more money in the very long run, but in the short term it helps them control their *cash burn*, or the rate at which they spend their investors’ money. Spending \$10,000 a month on office rent lets you spread the expense out, versus spending \$1 million all at once to build a new building.

Tax laws can also drive a company’s approach to OpEx and CapEx. For example, in the United States the Internal Revenue Service (the federal tax agency) requires companies to *depreciate* durable goods. Computer equipment is generally depreciated for five years, for example, and office furniture for ten years. That means if you spend \$100,000 on computer equipment—a capital expense—you can’t write that all off on the company’s taxes in the first year. Instead, you write off 1/5th of that amount, or \$20,000, on your taxes for each of five years.

Problem is, computers don’t always actually last for five years, so you’re stuck depreciating a piece of obsolete equipment. I worked for one company that got around the problem by *leasing* their laptop and desktop computers for three years—a far more realistic time frame for the equipment to remain usable. That converted the CapEx into an OpEx, since rent is an operational expense. When the lease was up, the manufacturer refurbished the machines and sold them to someone else, and brought my company a semi truck filled with brand-new computers on a new three-year lease.

20.4.2 Driving business decisions

Understanding how OpEx and CapEx work, and how your company feels about them, can help explain a great many financial decisions. For example, it might seem weird to have your CEO tell you that the company is going to slow down hiring to preserve cash, and then turn around and announce an \$80 million acquisition of another company. What gives?

Payroll is an operational expense. Broadly speaking, companies don't take out loans to pay for operational expenses. That's because OpEx doesn't increase the company's value. If you have a \$1 million annual payroll, and you take out a loan to pay that, you might wind up paying \$1,250,000 by the time you factor in interest on the loan. So you paid 25% more than your actual payroll, and didn't increase the value of the company at all. Your investors will not be amused.

Mergers and acquisitions are usually capital expenses. Rather than spending on-hand cash, you could take out a loan for that \$80 million, and pay it off over a couple of decades. Investors are fine with that, because you're increasing the value of the company by \$80 million or more at the same time. The actual \$80 million is a "wash," in other words; you're turning cash into a new non-cash asset. The operational expense involved—the interest on the loan—is the "price of doing business" to increase the company's value, and you'd normally expect to see the company use the new acquisition to increase overall company revenue, thus offsetting the operational cost of the acquisition.

Today, especially with regard to technology expenses, most companies are moving as much as they can to OpEx—meaning they're hosting software in the cloud, rather than on their own data centers. That can let them lower other OpEx (like the salaries of the people they paid to maintain their own data center), and potentially convert CapEx to cash (like selling the building the data center was in).

OpEx can sometimes also enable a company to scale faster and more granularly. For example, one dot-com I worked with ran its entire website on three servers that we owned. When we were about to get a big marketing push on a daytime television show, our CEO asked if the server "farm" could handle the load. "No way," we told him. He asked if we could increase the site's capacity. "Sure," we said, "but it's going to cost like \$40,000 in hardware and take a month to put in place." We didn't have the \$40,000 to spend, unfortunately—that's a big CapEx for a small startup. Today, we'd just scale out our Azure or AWS assets by pushing a button, pay a little bit of additional OpEx for the duration of the increased load, and then scale back when things die down, reducing our OpEx.

20.5 Business architecture

Business architecture refers to how a business is structured to get its work done. This may seem a bit esoteric, but I find it fascinating, as well as helpful in understanding the internal framework of a company and how it drives a company's behavior. There are many different ways to express this architecture, but my personal favorite is the approach that uses the concepts of *functions*, *services*, and *capabilities*. Many people use these terms, along with others, as if they're interchangeable, but in the real deep-thinking world of business design, they each have a distinct and vital meaning.

Perhaps the most granular level is a *capability*. A capability is a set of tasks that someone, or some group, within the company can do. Capabilities can include things like loading boxes onto a truck, processing the data needed to underwrite a loan,

booking a hotel room, and so on. On their own, most capabilities aren't things you could sell directly. For instance, being able to make a good hamburger is an excellent capability, but you can't make a restaurant out of that one capability. You also need to be able to take an order for one, accept payment for it, clean the table when the customer leaves, and so on; those are all distinct capabilities.

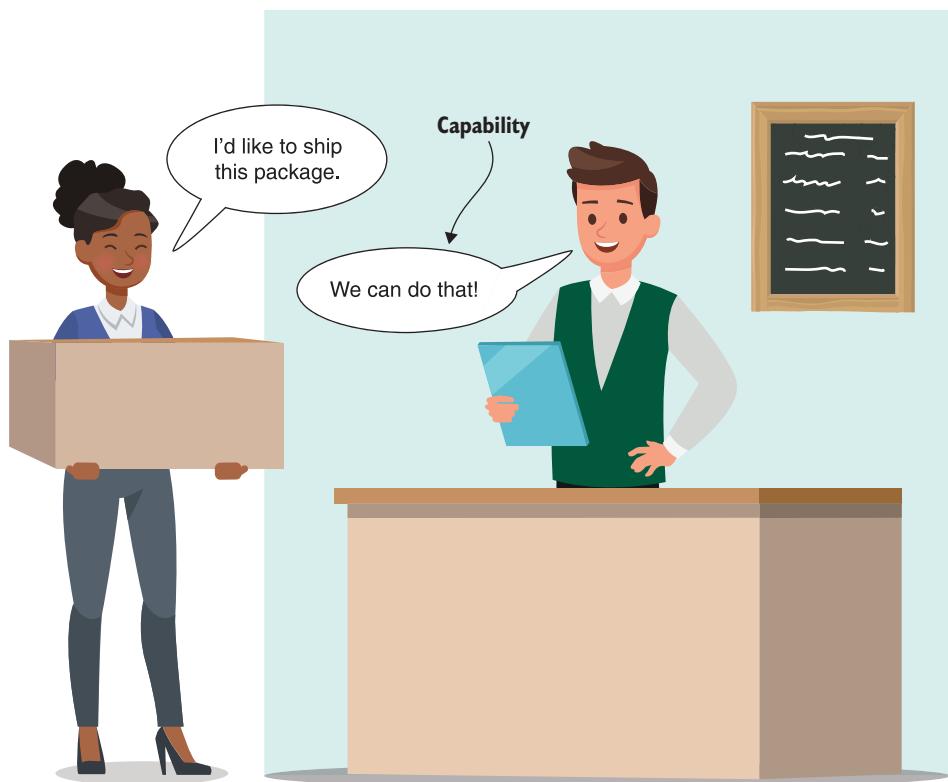


Figure 20.2 Business capabilities are the granular things a business can do.

A business function could be defined as an outcome, such as an element of a product or a service, which a customer might want. Consider a fast food menu: it might include an item like a burger, which a customer wants. But the burger gets broken down a bit with options that the customer cares about: meat temperature, toppings, bun type, and so on. Those options are functions. In a more business-y example, consider a customer that wants to ship a package. They take it to a shipper, and “shipping a package” is a capability of that shipper. Within that capability are functions like shipping speed, ability to go to specific destinations, ability to handle certain package sizes, and so on. You can't just have a business that offers “ketchup as a topping”: that topping is merely a function of a business capability.

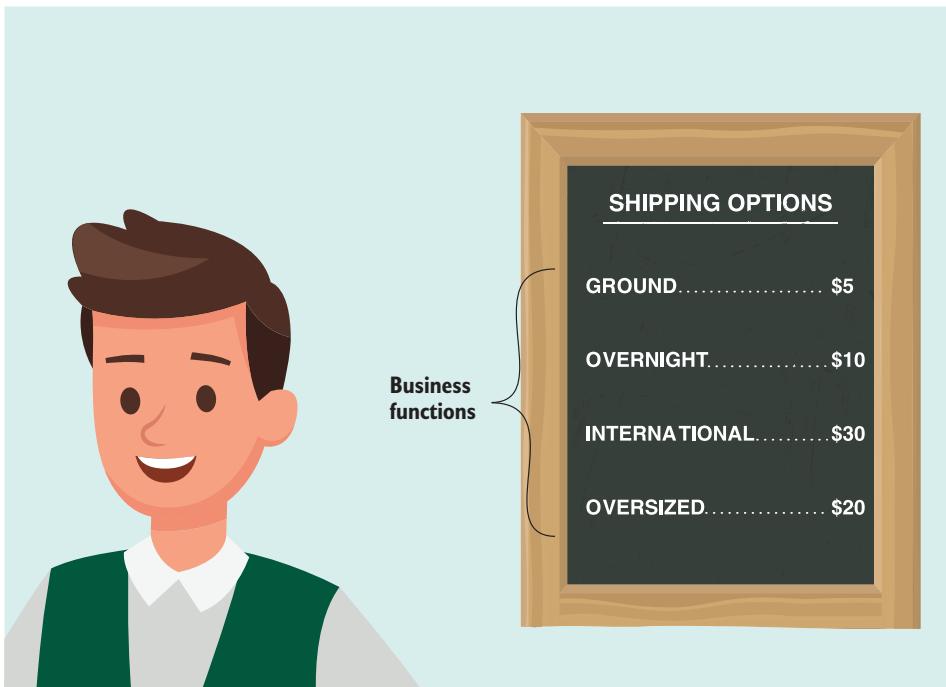


Figure 20.3 Business functions are outcomes that a customer might want.

Internally, *services* combine capabilities, usually in a specific order as defined by a *process*, to help fulfill the promise the functions have made to customers. Shipping a domestic package is a service: it requires specific capabilities, like calculating a cost, selecting a form of transportation, planning the logistics, and executing the actual movement of packages, and it requires that you perform those capabilities in a particular order.

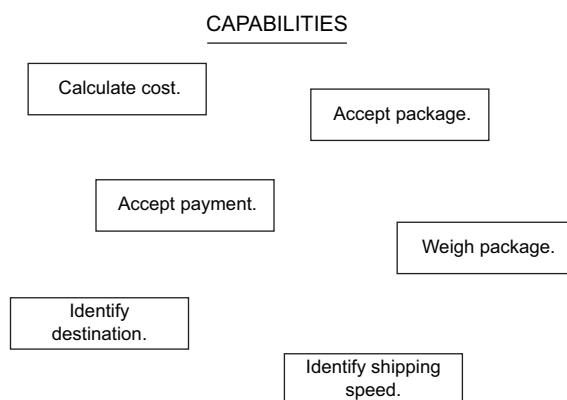


Figure 20.4 Business services are collections of capabilities.

The business *process* defines the order in which those capabilities happen. So capabilities, executed in a specific process, create a service; services usually correspond to functions so that when a customer orders a function, the service executes and the customer is served.

It is also possible, and quite common, for higher-level processes to exist, which themselves consist of distinct services that are executed in a particular sequence. A service, in this case, is some standalone, self-contained set of capabilities that may be called upon by multiple processes. For example, a “clean-up” service might be called upon by multiple different processes, including processes like “respond to spilled beverage on aisle 4” and “respond to severed limb on the loading dock.” Those processes might include other services as well, such as “call an ambulance.”

So a process can consist of *more than one service*, and each service can have, within it, its *own* processes for accomplishing that specific service. Functions—because they’re an abstract, customer-facing definition of something the business offers—can often serve as a clue to what an organization’s org chart should look like. I’ve often said that customers should be able to infer at least the top layers of a business’ org chart simply based on what the business appears to sell. In a Las Vegas casino-hotel, for example, I would expect top-level org chart divisions to include things like Hotel Operations, Gaming, Entertainment, Food & Beverage, and so on. If I were to find a resort whose org chart didn’t basically line up with that, then I’d expect that resort to operate inefficiently and to have a difficult time providing customers with a great product. If your functions—your menu of offerings—are what you sell, then almost by definition your org chart should be designed to facilitate the delivery of those corresponding services. If you want the short version:

- Functions are an abstract, customer-facing definition. The only thing they call upon are processes.
- Processes combine services in a specific sequence to deliver on a function.
- Services combine capabilities, usually executed according to their own internal process.
- Capabilities are the granular, individual tasks that, collectively, make things happen.

All of this implies a level of modularity. A well-designed service can be used across high-level business processes. For example, if your company has four different departments that ship things to customers, they should ideally all be using the same internal service

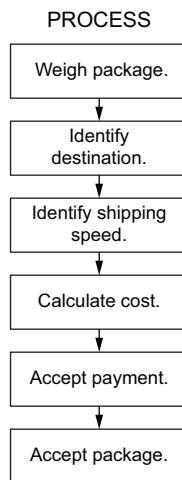


Figure 20.5 A business process assembles capabilities into a sequence that leads to an outcome.

to do so. If they're not, you're likely less efficient than you could be, and you're probably wasting time, money, and effort, as well as creating inconsistent outcomes for customers. *That* is why “business architecture” is a thing: analyzing a business to understand where unnecessary service duplication exists (which happens all the time in companies that have grown organically) and deliberately architecting the situation into a more efficient and consistent model.

I fully acknowledge you'll see other definitions for these terms, but by and large, the most common ones align roughly with what I've described here. Even if your business uses these terms or others somewhat differently, just understanding that there *is* a structure to these things can make it easier to learn the structure of *your* business. The “rightness” or “wrongness” of how I've used these terms should be less critical to you personally than knowing that they *are* essential terms that relate to essential concepts of business.

So what's all this mean to you? If I make pizza for a pizza shop, I should understand actually how my *capability* contributes to the *services* my business executes, and I should understand the *processes* in which my capabilities are performed. I should understand how the corresponding *functions* are presented to our customers. I should, in other words, start by reading the menu customers see. I should understand the process of how the kitchen works. All of that information helps me to understand the full context of my job and helps me make sure I can perform it smoothly. I should understand that changing how I perform my capability will impact other people who perform other capabilities within the same processes that I do: sticking a pizza in the oven at the wrong time is going to disrupt the entire down-chain process, resulting in a lousy delivery to the customer. My pizza-making service, combining as it does capabilities like dough-making, topping-applying, and cooking, is a service that should be consistently usable by multiple processes, such as “make pizzas to sell by the slice,” “make pizzas for dine-in customers,” and “make pizzas for delivery.” The function “order a pizza,” which is what customers see, can then call on any of those processes as appropriate, knowing that the outcome will be consistent.

20.6 Further reading

- *Business Mathematics*, 12th ed., Gary Clendenen, Stanley A. Salzman, and Charles D. Miller (Pearson, 2018)
- *How to Lie with Statistics*, by Darrell Huff (W. W. Norton & Company, 1993)

20.7 Action items

- Calculate the hourly cost of yourself in fully loaded terms. Ask your manager, or even your company's Finance team, what multiplier they typically use to calculate fully loaded salaries.
- Can you think of an activity that you and your team perform on a regular basis that is overly cost-consuming? That might be working with a difficult source-control system, unnecessarily rebuilding servers, troubleshooting DNS problems,

or some other technical activity. How much would be saved per year, based on the fully loaded salaries of the individuals involved, if those activities could be made to consume less time?

- Thinking about functions, processes, services, and capabilities, what is your team responsible for? What about the larger department that contains your team? What about your company as a whole? It can often be helpful to understand how a company's services, for example, boil down to the functions of your particular team. That understanding can make it easier to understand your specific impact on company-level outcomes.
- What are some of the OpEx and CapEx expenses your company experiences on a regular basis? (Hint: payroll is an OpEx.) What CapEx expenses could be shifted to OpEx, if the company had a desire to do so?

21

Tools for the modern job hunt

The modern job hunt is a lot more complex and nuanced than our school teachers taught us. We’re dealing with artificially intelligent screening systems, keyword-driven applicant tracking systems, more complex compensation packages, and a range of other factors that are often invisible to job-seekers. It’s time to modernize your toolbox, because a successful job hunt starts months—or even *years*—before you actually start applying.

21.1 **Job hunt tasks to do now**

You can’t wait until you’re about to engage in a job hunt to actually *start* the job hunt. **There are two specific tasks you need to do well ahead of time, and continue to work on all the time. Those are**

- Manage your brand, and continue managing it on an ongoing basis. I discuss how to do this in chapter 2.
- Create, nurture, grow, and engage your professional network. I discuss this in chapter 3.

I have to be honest with you: while this chapter includes a sizable section on getting your résumé or CV in order, it is your network that is more likely to get that résumé or CV in front of a hiring manager. You cannot rely on simply submitting your résumé to the many online job-posting websites. Those sites can, for any given job posting, generate thousands of applicants. Your network is what will likely get your résumé pulled out of the pile and in front of human eyeballs. **Yet it can take years to build a supportive, engaged, and meaningful professional network**, which can include colleagues, recruiters, co-workers, and others that extend your “reach” into

the world at large. If you don't start now, and continue throughout your career, you will have a harder time moving your career in the direction you want to take it.

21.2 **Review your brand**

In chapter 2, I explained the concept of "Brand You," the public-facing version of you that tells others what to expect from you before they've even met you. At no time is "Brand You" more important than when you're preparing for the job hunt.

To be clear, the time to work on "Brand You" is not when you're ready to look for a job; your brand is something you need to be aware of, and nurturing, constantly. But the job hunt is a very good time to sit down and seriously review the state of your brand, and start to make any tweaks needed to help ensure a successful job hunt.

21.2.1 **Your professional brand and the job hunt**

Think about the job hunt process from the hiring manager's point of view. They have the seemingly impossible task of meeting a complete stranger, and based almost entirely on things that stranger has written about themselves (their résumé) and said about themselves (the interview), the hiring manager has to decide if the person is qualified for the job, a good fit for the team, and a smart hire.

I've been a hiring manager lots of times. It's terrifying. You're making this huge decision that will impact you and whatever you're accountable for. It'll impact the team you already have in place. It'll impact the entire company. If you make a bad choice, you're left in the even-more-horrible situation of possibly having to let someone go. Even thinking about all that makes my stomach clench up.

And so in our modern, always-connected, always-on era, the hiring manager turns to the internet, in hopes of finding out more about the *authentic* you. The real you. Not the "you" that you bring to the interview, or the "you" you created a résumé for. They want to get to know the *genuine* you. And what better place to do that than in the place where most people let down their guard and act like their true selves? That's right: the internet. Specifically, social media. And, for technology professionals, "social media" means a lot more than Facebook, Twitter, LinkedIn, and Instagram. A *lot* more. Your "social footprint" can include sites like GitHub, StackOverflow, Q&A websites, and other tech-specific areas where you can be seen, and where your contributions can be reviewed. Everything that you put out into the public becomes part of your *brand*. It's a representation of who you are, and so you want it to be your best face.

But I want to stress a word, here: *public*. Very few of us are exactly the same person at work as we are in private. In private, I curse a lot—I was raised in and around the US Navy, and certain impolite words are standard verb/noun/adjective/adverb in everyday conversation. But that's not appropriate for work. So when it comes to your professional brand, we're focusing on your public footprint.

21.2.2 Reviewing your public footprint

Let me be blunt right up front: not everything I do online is public. My Facebook activity, for example, is purely private, just for close friends and family. And my profile is locked down accordingly: if you’re not on that friends-and-family list, you don’t see anything. I verify that by using a different web browser, on a different computer, to confirm what the public can and cannot see.

And that’s your first step: decide what’s public, lock down everything else, and verify that the lockdown did what you wanted it to do. As a professional, I’m sure you’d never get rip-roaring drunk at a company function and start dancing on desktops. Make sure that your public-facing brand doesn’t imply anything different: keep those party photos *private*.

And as you’re reviewing what you make public, give some serious consideration to the “friends” who can see the private side of your life. I have numerous colleagues I’m friendly with; *very* few—like, two—are “Facebook friends.” Facebook is my private life; the minute I let someone from work into that life, it ceases to be private and almost inevitably becomes part of my professional life. So I draw a pretty hard line.

INVENTORY YOUR FOOTPRINT Take an inventory of all the places you can be “seen” online. Which of those are you designating as “private life,” and which are allowed to become part of your professional brand? And make sure you’re focusing on your entire online presence. For example, one easy-to-overlook area is online Q&A websites, such as discussion forums or sites like StackOverflow.com. Are you the type of person who asks a lot of questions, but never answers? That gives me a certain impression of you as a professional, and I might worry that you’d be the same way at work, creating a lopsided team balance. Do you post well-written questions and answers? If not, I might worry about your communications skills, which I guarantee are an important criterion for any hiring manager. If you do post answers, are they polite, concise, and supportive? Or are they snarky and sarcastic? I know which type of person I’d rather have working on my team!

For the aspects of my online life that are part of my professional brand, I’m very careful about what goes there. My GitHub account, for example, includes a number of open source technology e-books, some open source software projects, and other things that I’m *proud* for a potential employer to see.

My Twitter account activity largely reflects my professional values. For example, I obviously have political opinions—everyone does. But I don’t express those opinions as part of my professional brand, because my political opinions aren’t part of my professional life. When I’m working in the office, I keep my political opinions to myself, because I feel they don’t contribute to the job I’m paid to do.

And that’s another way to think about your public footprint: anything you put online that can be accessed by the public should be things you’d also be comfortable taking into work and either saying or pinning to the wall—with your name attached.

I had a colleague who was on a job hunt one time, and I gave him all of this advice about his public internet footprint. He claimed to have followed it, but the job he was really longing for wound up declining to hire him. I knew a lot of folks at that company, and so I asked if there was anything *unofficial* I could tell him. Unofficially, they'd taken a look at his Twitter account, and become alarmed at some of the things he'd retweeted. They were concerned that his retweets represented his own opinions, and that he'd feel comfortable bringing those opinions into the workplace—which *they* weren't comfortable with.

I have a mantra for the folks who attend my workshops, when we start talking about brand and public footprint: Lock it down, Clean it up, Shut it off, Show it off.

- *Lock it down.* Anything that's part of your private life needs to be private. Lock it down, and confirm that lockdown. Be careful about who you let into your private life; if you wouldn't repeatedly invite a co-worker over to your home, then don't make them part of your private online presence.
- *Clean it up.* For anything remaining in the public, make sure it reflects who an employer can expect you to be at work, in an office with your colleagues. If a continual stream of lolcatz memes wouldn't be appreciated in the office, then keep it out of your public footprint.
- *Shut it off.* If part of your online presence isn't contributing to your professional brand, and locking it down isn't an option, then shut it off. Close the account, delete everything, and walk away.
- *Show it off.* For the parts of your online presence that *do* contribute positively to your professional brand, *show it off!* Community contributions, blog articles, open source projects—all of those things deserve the attention of a prospective employer, and you should make sure they're something you refer to in your résumé and other communications.

Everything you do online contributes to your professional brand, unless you've locked it down so that it's fully private. Make sure that professional brand is what you *want* it to be.

21.2.3 How would others describe your brand?

Once you feel you've gotten your online presence fully aligned with the professional brand you want to project, ask others to describe your brand. I'll obviously include people I work with, or that I've worked with in the past. They've experienced me personally, and have seen a lot more of "Brand Me" on a day-to-day basis than anyone else. But I'll also include online colleagues: people I've interacted with on Twitter, on Q&A forums, in GitHub repositories, and so on. I'll ask them to describe how they see me as a technology professional.

Sometimes, I'll ask for all this feedback by means of anonymous surveys, which I set up using free online survey websites. I'll ask just a few basic questions, such as

- Do I seem like the type of person you'd want to hire or work alongside?
- Would you have concerns about me being on a team with you?

- Do I come across as a valuable technology professional who contributes to positive team outcomes?
- Do I seem like someone who could make a team more effective or efficient?

I'll also leave room for respondents to leave comments, and I try to take those in the spirit in which they're offered. It's sometimes hard to not get hurt or offended when the comments aren't all sunshine and roses, but I take them as an opportunity to reflect. I often don't realize how I'm being perceived by others, and so while I don't intend to come across negatively, it's good to know when I'm doing so unintentionally. It gives me a chance to think about it and, if I choose, to change things up a bit.

21.3 Update your résumé

In the United States, we call it a *résumé* (or *resumé*, if you're being fancy). My European friends call it a *CV*, for *curriculum vitae*. Whatever you call it, when you're on the job hunt it's one of the most important documents you'll have.

Résumé-writing is a cross between fine art, science, and a little bit of magic. Entire books the size of this one have been written on the subject; in this short chapter I obviously won't have room to condense the entire body of human knowledge on the subject. What I want to do instead is line you up with some of the most important and often-overlooked aspects of a résumé.

THE TIME TO WORK ON YOUR RÉSUMÉ IS BEFORE YOU NEED IT Even if you're not on a job hunt right now, please read this chapter. Putting together a great résumé is hard and time-consuming; working on it *before* you need it is the best way to keep it ready for when you do.

21.3.1 Résumé rules

The rules for what should go into a résumé have changed *drastically* since I won first place in my state's Vocational Industrial Club of America (VICA) Job Interview Competition (yes, that's a thing). And if you haven't written a résumé in a while, you might be surprised at some of the modern "best practices." Here are a few of the old rules that no longer apply:

- *Maximum one page, both sides.* In our modern era of artificial intelligence, Applicant Tracking Systems (ATSS), the length of your résumé is often less important than its contents. That doesn't mean you should go overboard, but do feel free to include everything of value. Just keep in mind that at some point a human may read your résumé, and you don't want them falling asleep halfway through.
- *Begin by stating your career objective.* This is a sure-fire way to make your résumé look outdated. Think about it: you're applying for a job, which means you're applying to solve a problem for an employer. Yet you start your personal sales pitch by talking about yourself? Skip that, and get to something your prospective employer will care about: what you can do for them.

- *Only use bullet points.* There's a lot to say for a nice, concise bullet list, but if your résumé looks like you've been splatting flies on it, it can make a person's eyes glaze over. I'll cover alternate approaches later in this chapter.
- *Keep it impersonal.* Half the interview process is the prospective employer trying to get to know you; you can jumpstart that process by letting some of your charming personality leak into the résumé. Résumés that read like they were written by a robot can be boring. Don't get carried away—you don't want to come off like you're not serious—but work to make your résumé look like it was written by a human being. I like to read mine back to myself out loud. If it sounds like something I'd say in person, I know I've got the right balance.
- *Offer references.* A prospective employer will ask for this *if* they want it; many won't, and you shouldn't offer it proactively. Most references are seen as biased toward you, and so they're often not given a lot of weight.
- *Include your mailing address.* Mail? These days? Nope. Include a personal phone number and email address, and you'll be fine.
- *List All The Things.* I'll detail this more later in this chapter, but don't feel that you need to list skills or competencies that are totally base-line for the position you're applying for. For example, if you're trying to get a job as a Senior Software Engineer, you can probably skip your skills with Microsoft Word and just get to the good stuff.

IT'S ABOUT THE JOB The point of a résumé is to secure an interview for the job you want. Look closely at your résumé and delete every word and punctuation mark that doesn't help move you toward that goal.

21.3.2 Starting your résumé

I use LinkedIn to start my résumé. My job-searching role model, Jason Alba of Jibber-Jabbing.com, put me onto this approach, and I like it. LinkedIn gives me a place to enter my work and education history, highlight key skills (and it puts a limit on how many you can have, which forces me to pick the most relevant ones), and it lets me include things like awards, certifications, and other achievements. As a bonus, that LinkedIn profile is one of the ways recruiters and potential employers can find me, so it does double duty.

As a bonus, LinkedIn will even export your profile as a formatted résumé, although as I'll discuss later, you should really consider if that's your best option. At the very least, my LinkedIn profile is a place for me to keep all of my information on file for quick reference when needed. You're welcome to check out my profile at <https://www.linkedin.com/in/concentrateddon>. I will highlight a couple of things about my profile that aren't helping it become a better résumé, and explain why I went the direction I did:

- A lot of my work experiences, if described in detail, would be too long for a résumé—a 6-page résumé is just too much!. However, I include longer descriptions in my LinkedIn profile because I want as much as possible in my profile. If

I apply for a position, I trim this down on the résumé to whatever is appropriate for the position.

- I have a lot of experience outside of traditional work positions—like my non-profit, and my work as a freelance author—that might not make it onto an actual résumé. That’s because those things aren’t always relevant to a job I’d be applying for. But I include them on LinkedIn to make my profile represent a fuller view of who I am.
- My “Skills” section is a little bit of a hodgepodge, in part because I tend to experiment with it to see what LinkedIn does in response. I use this section as inspiration when writing an actual résumé, but I don’t rely on copying it verbatim.

It’s fine if you don’t like using LinkedIn. You can do something similar in a word processing doc, in a database, or whatever is most comfortable for you. The point is to have as much résumé content as possible at your fingertips for when you need it.

One of the best pieces of advice I ever got was to continually update my résumé. Today, I take that to mean continually updating my résumé database, which for me is my LinkedIn profile. If I do something significant at work—launch a new product, create a major financial savings, take on a new responsibility, or whatever—I update my profile. That way, if I ever need to write a résumé, all those facts are right in front of me. If I don’t update as I go, then it becomes too easy to forget something important when the time comes to write a résumé.

21.3.3 Every résumé is unique

While I used LinkedIn as my source material for résumés, I do not just export my LinkedIn profile into a résumé. That’s because every job I apply for gets a unique résumé that has been tailored for that job. I’ll sometimes tell people that they need a job-specific résumé for every job they apply for, and the response is often a groan and an eye-roll. I get it: that’s a lot of work. But . . . do you really want the job?

Let me put it another way: Do you know why Ford makes so many different vehicles? Because not every vehicle is perfect for every situation. Some vehicles are great at ferrying half the soccer team to pizza after the game. Some vehicles get amazing gas mileage. Some have great cargo capacity, and some are more luxurious. Ford tries to address many different market segments, and they can’t do that with a “one size fits all” vehicle.

So why should one résumé be a perfect fit for every job you might apply for? When you tailor a résumé to the job you’re applying for, you’re acknowledging that your potential employer is unique. They have specific needs, and you’re prepared to help meet those needs. You’re taking an interest in them, rather than forcing them to “translate” your résumé into something that might be useful to them. In other words, you’re doing some of their work for them, which is exactly what employment is all about. If you can’t be bothered to tailor your résumé, what does that say about the kind of employee you’ll be?

21.3.4 Analyze the job posting

For the next few sections, I'll use a specific, real-life job description to show you how I analyze and interpret job postings . However, because job listings come and go all the time, I'm going to direct you to my own website: <https://donjones.com/example-job-description>, where I'll update this example occasionally. That way, you can follow along and we can both work from the same example. I suggest you copy and paste the job description into a text editor, so that you can highlight key bits as we go. Also, just for your convenience, I'll include the relevant bits of the job description here:

Due to our continued growth, we are adding a talented, highly motivated and service delivery-oriented DevOps Engineer to our team. This integral role will focus on delivering end-to-end product solutions where stability and security are the highest priorities.

The DevOps Engineer will be an integral part of our Infrastructure and Security team. This key team member is responsible for deployment of infrastructure as code with expertise in AWS. Kubernetes experience is required! We have an open, visible environment and this position will interact with decision makers on a daily basis. We align with a tech firm in that we are big on collaboration, making decisions quickly, and remaining as dynamic and agile as possible. Our roles are challenging and rewarding. If you like a fast-paced environment where you are staying at the forefront of technology and constantly learning, this position would be a great fit.

Our ideal candidate for this job (is):

- 4+ years of DevOps experience—supporting production cloud infrastructure (must include AWS), containerizing modern web applications (Docker, Kubernetes), and the cloud native ecosystem
- 4+ years of experience with Linux-based server operating systems
- 4+ years of experience with Infrastructure-as-Code and Configuration-as-Code (Terraform)
- Strong understanding of web service fundamentals, such as HTTP/S, SSL/TLS, TCP/UDP, performance monitoring, caching, load balancing and logging
- Strong understanding of and experience maintaining a healthy CI/CD pipeline, as well as familiarity with deployment strategies
- Working knowledge of relational DBMS such as MySQL
- Proficient in a modern programming language such as Go, Python, or Rust
- Experience with a service mesh technology (Istio, Consul)
- Experience working with software and SaaS vendors through trials, new implementations, renewals, and decommissioning
- Security-driven, highly motivated, self-starter who thrives in a fast-paced, highly technical environment
- Effective collaborator, experienced in creating technical partnerships across teams to achieve a shared mission
- Experience using agile methodologies to plan and track project work

- Excellent written and verbal communication skills and ability to communicate project plans and progress with stakeholders across the business, including experience producing technical documentation
- Enjoys new technological challenges and is motivated to solve them
- This position is part of a distributed team based out of our Tampa, FL office, which requires the ability to effectively collaborate with colleagues remotely.
- Must speak fluent English and be located within the United States

Also preferred:

- Certified Kubernetes Administrator (CKA) or be working towards it
- AWS Certified Architect or be working towards it

The first thing I do with a job posting like this is make a list of

- Requirements
- Preferences
- Keywords

IDENTIFYING HARD REQUIREMENTS

When I refer to *requirements*, I'm talking about objective, non-negotiable items. This posting says that the "ideal candidate" has "4+ years of DevOps experience," but that's not a *requirement*. The word "ideal" means there's some room for negotiation. For example, perhaps I only have 3 years of DevOps experience, but I did it in a high-pressure, fast-moving environment that was in the exact same industry as this company. They might be happy with that.

So for what I call "hard requirements," meaning non-negotiable, objective requirements, I found

- Kubernetes experience
- Must speak fluent English and be located within the United States.

That's right, that's the only item this particular posting has that uses the word "Must," and one that says "required!" Everything else is probably negotiable to some degree or another. Looking for these "hard" requirements is important because it can help you overcome your own fear or shyness around a particular job posting. This posting lists a lot of criteria, but no employer expects to get every single one of those in one job candidate. Only the "hard" requirements are probably non-negotiable; if you can meet those, then it's worth digging into the rest of the posting in more detail.

Be careful when parsing the language in a job posting. For example, the infamous "Must have a four-year degree or equivalent work experience" is not a "hard" requirement. What is "equivalent work experience?" Unless the job posting specifies, that's a negotiable point. If you have zero work experience and no degree, you probably aren't what they're looking for. If you have some work experience and no degree, consider that to be a "preference."

IDENTIFYING PREFERENCES

My example job description has lots of preferences. I tend to rewrite these into a checklist for myself. I break that checklist down into two sections: the first section is the items that are more objective and easily provable, and the second section is the items that are more vague or subjective. If I run across items that seem *especially* vague or subjective, I'll put those into a third section:

Objective preferences:

- 4+ years of DevOps experience
- Experience with Amazon Web Services (AWS)
- Experience with Docker and/or Kubernetes
- 4+ years of Linux experience
- 4+ years of experience with Terraform (or possibly alternate solutions)
- Experience working on a trial or new implementation

Subjective preferences:

- Strong understanding of HTTPS, performance monitoring, caching, load balancing, and logging
- Strong understanding of CI/CD pipelines
- Working knowledge of MySQL or an alternate relational database management system
- Proficient in a language such as Go, Python, or Rust
- Experience with Istio and/or Consul
- Experience using Agile methodologies

Very subjective preferences:

- Experience working with software and SaaS vendors
- Security-driven
- Highly motivated
- Effective collaborator
- Works across teams
- Written and verbal communication skills
- Enjoys new technological challenges

I'll offer you something from my own experience as a hiring manager: that third section is often the one I care the most about. It's the items that are hardest to analyze in a résumé, the hardest to pick up in an interview, and the ones that often make me the most anxious about the hiring process. That means, as a résumé *writer*, you need to focus a solid amount of effort in communicating those things.

The first section is easy to address in a résumé: either you have those things or you don't. If you have them, list them. The middle section is more subjective because it doesn't say how much experience or proficiency the company is looking for. The second

section's items are probably of strong, but not primary, importance. The company is going to be looking for these things, but they may even expect some of them to be "baseline" skills for the position. I want to make sure my résumé highlights whichever ones of these I have, and I probably need to put some writing effort into communicating the nature of my experience.

IDENTIFYING KEYWORDS

In a world driven by machine learning, keywords are your friend, and they're a big reason why you need to make a unique résumé for each job you apply for. **Remember, hiring managers won't see your résumé until you pass the machine learning filters, and those filters rely heavily on keywords.**

Your résumé cannot simply have a bullet list of keywords; the robots that will first read your résumé are usually too smart for that. You want to make sure the keywords appear organically throughout the work experience on your résumé. My checklist of keywords for this job posting:

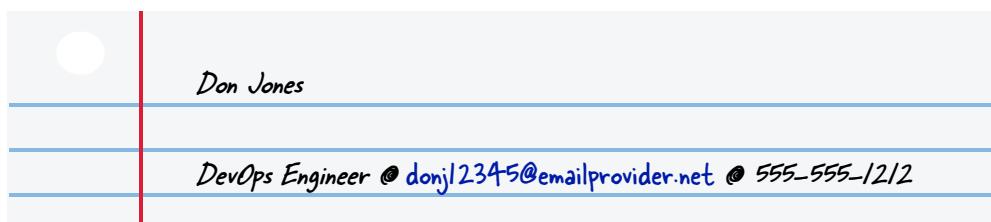
- AWS
- Docker, Kubernetes
- Linux
- Infrastructure-as-Code (IaaS)
- Configuration-as-Code
- Terraform
- HTTP/S
- SSL/TLS
- TCP/UDP
- Performance monitoring
- Caching
- Load balancing
- Logging
- CI/CD pipelines
- Relational DBMS, RDBMS
- MySQL
- Go, Python, or Rust
- Istio, Consul
- SaaS vendor
- Security
- Collaborator
- Partnerships across teams
- Agile
- Written communication skills
- Verbal communication skills
- Technical documentation

Why those terms? They're all the names of products, tools, technologies, or frameworks that were mentioned, or the commonly used phrases to refer to specific "soft skills" (like communications). These are the words that, to someone who *is* qualified for this job, really stand out as meaningful.

Now I know that my résumé isn't ready until every one of those terms that I legitimately *can* include *are* included. And that's where I turn to my LinkedIn profile (or whatever you're using to track your "résumé source material"). You know, I did have that one job where I was helping build CI/CD pipelines, but I didn't use that term specifically. So I'll update LinkedIn to include the new term, and that'll be part of what I pull into the new résumé. That's one of the reasons my LinkedIn profile is a little lengthy: it contains a lot of the keywords from past job applications.

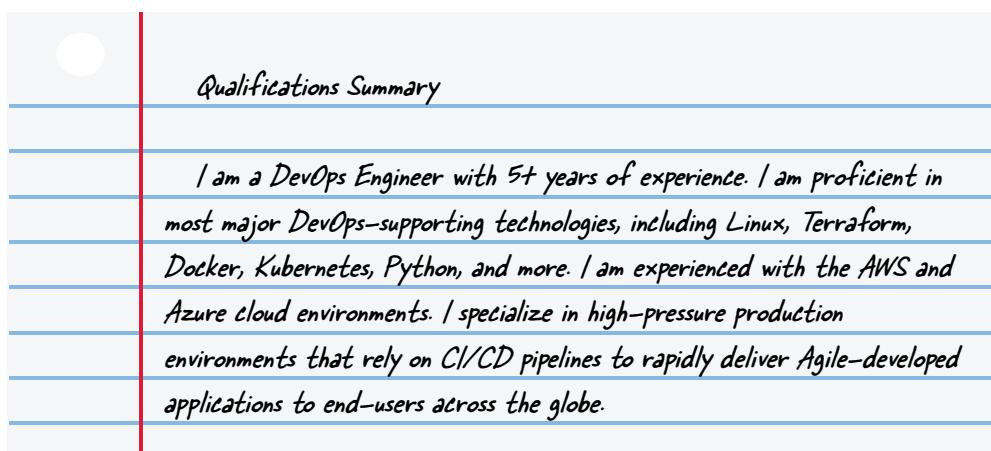
21.3.5 Writing your résumé

Always be sure to start with your name, personal phone number, and an email address that you've set up for the purposes of job interviewing (I used to simply use my personal email address, but you end up getting added to lots of lists and I now use a separate address):



QUALIFICATIONS SUMMARY

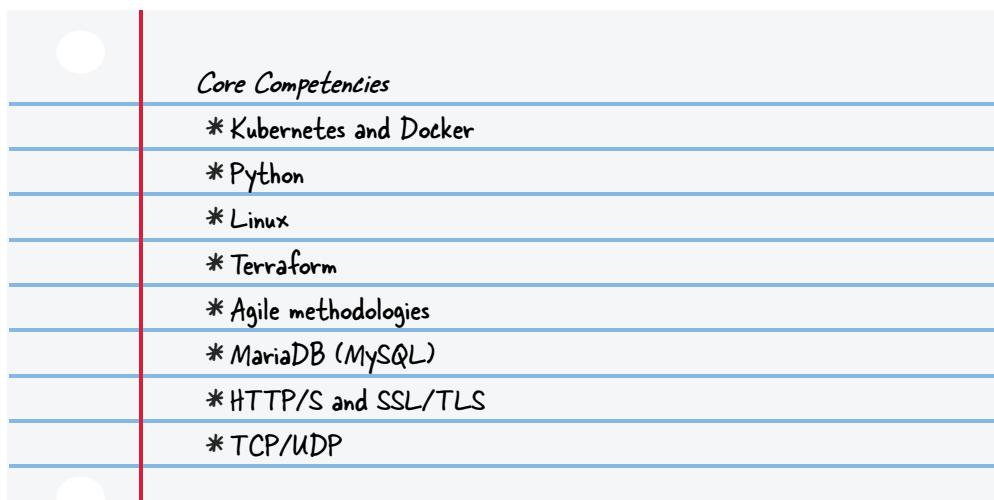
Today's résumés should then feature a *qualifications summary*. For example:



That is nailing down some of the most critical keywords from the job posting, and putting a tiny bit of context around them. Think of this as your “teaser statement,” written to entice the hiring manager (and machine learning algorithm) to keep reading.

CORE COMPETENCIES

Follow the qualifications summary with a section titled “Key Skills,” “Core Competencies,” or something similar—whatever is most comfortable-sounding for you personally. This section should list no more than about a dozen skills that are applicable to the job posting, which you feel you are proficient in. *Proficient*, not World Class Expert; that means you can hold down a job that requires you to use, and learn more about, these skills.



Notice that I listed “MariaDB (MySQL).” The job posting calls for MySQL, but my experience is actually with MariaDB. MySQL and MariaDB are functionally equivalent, but this allows me to be truthful about where my experience lies, and still snag the keyword that the machine will be looking for.

WORK EXPERIENCE

Next up is your work experience, in reverse chronological order. If you’ve been in the industry for a while, you don’t need to go back to your first high school job flipping burgers—as with everything else on the résumé, *stick with what is relevant to the job posting*. If you have work experience that in no way contributes to what the job posting is after, list it so as to not have a gap in your history, but don’t waste space going into detail.

Remember that bullet points are your friend, and this is where you want to capture the job posting’s keywords in the context of your experience. This is also where you get to brag about specific accomplishments, especially when those accomplishments speak to a requirement of the job posting.

DevOps Engineer @ Startup.com @ Jan 2/18 – Jul 2/20

- * Technical lead for team managing over 3,000 Kubernetes deployments monthly
- * Built and maintained CI/CD pipeline including Team City and Terraform
- * Used Python automation to reduce manual deployment effort by 92%
- * Rebuilt AWS-based pipelines to realize 35% cost savings over 18 months
- * Built cross-team partnerships to better create and deploy technical documentation on pipeline architecture and usage

Notice how some of these bullets combine keywords with context: I've done Kubernetes, and I did it 3,000 times a month. I've got that "cross-team partnerships" and "technical documentation" keywords, but I've put them into context so a human can tell exactly how those keywords apply to my experience. And I've added a couple of numeric-based metrics that show the kind of improvements I can create. There's lots more I could say about my time at that job, but these are the things the company I'm applying to will care about, based on their job description.

COMMUNITY WORK AND ACHIEVEMENTS

Finally, your résumé should end with any relevant volunteer or community work you do, and with any awards, achievements, or accolades you've earned. Again, relevancy is key: don't list the time you spend working with the local Parent-Teacher Association unless it's relevant to the job posting.

Community Work and Achievements

- * Lead for the monthly Chicago Terraform User Group meetings
- * Recipient of AWS Community Contributor Award, 2020

This last section of your résumé is a bit like a college application. When I was in high school, we were taught to make sure we had something to put in *every* section of the college application. That meant some of my computer geek friends went out for field hockey, just so they'd have something to put in the "Sports" section. Others joined the debate team and the yearbook staff, so they'd have something for the "Extracurricular Activities" section.

When it comes to your résumé, you should make an effort to have community work to list, which means you're going to have to start doing that community work

well before you intend to need it for a résumé. That community work supports your career, even if your current job doesn't require it, and your career is what will get you that next job.

Community work can include things like helping run user groups, contributing to open source software projects, routinely answering questions in online Q&A forums, speaking at conferences, writing technical blog articles, and other contributions. These things might not be your personal passion, but just like the computer nerd who tries out for the baseball team, they're an important and visible part of your career.

FOCUS ON OUTCOMES, NOT EVENTS

I find that résumés often include bullets like this one:

- Launched cross-functional standards group to promote quality consistency across all project types

That's somewhat of a waste of space on a résumé. As someone reading that, I might think, "Okay, you launched it—then what happened?" Try to make sure everything on your résumé speaks to an *outcome*. That each bullet conveys *value* or *benefit*. If you can't reword a bullet to convey an outcome, drop it—it's not telling the résumé-reader anything useful about you. For example:

- Launched cross-functional standards group to promote quality consistency across all project types; after 6 months, this effort reduced project maintenance time by 26%.

Metrics—hard numbers that speak to business value—are the best. Consider these not-so-great bullets:

- Led team through two organization/company-wide re-orgs
- Rewrote and reformatted quality standards for document editors and reviewers to support learning effectiveness and scalability
- Created and launched system to support regular peer feedback

Now let's rewrite those with some metric-based value:

- Led team through two organization/company-wide re-orgs; maintained team turnover at 5% below company targets
- Rewrote and reformatted quality standards for document editors and reviewers to support effectiveness and scalability; increased team scale by 18% and effectiveness trended upward by 8%
- Created and launched system to support regular peer feedback; team NPS¹ increased by 5%

If you're thinking, "But wait, I don't have anything like a "team NPS" I can use as a metric," then I would say, "That's why you need to think about your résumé *before* you *need* your résumé." Start looking at the good you do in your organization, and implement ways to measure that good.

¹ Net Promoter Score

21.3.6 Formatting your résumé

This is a tough section to write, because formatting a résumé these days is a tough topic. On the one hand, you want a nice, modern-looking résumé that shows you're a modern, hip technology professional who cares about your appearance. On the other hand, your résumé is mostly going to be read by artificial intelligence algorithms that don't care what it looks like. Consider the example in figure 21.1, which is a sample of the kinds of beautiful résumés produced by the online service <https://cvonline.me>.



SHARON KIRKWOOD
Archaeologist

✉ sharonkirkwood@email.com
☎ +00 000 0000 0000 📍 Birmingham, England, UK

WORK EXPERIENCE

CURATOR 08/2018–Present
Birmingham Museum and Art Gallery

- Managed installation of major exhibits according to museum standards
- Scheduled and coordinated exhibitions, including tour guide hiring and training
- Coordinated marketing/publicity resources
- Served as managing editor of exhibit brochures

ARCHIVIST 11/2016–07/2018
Museum of London

- Supervised the indexing and archiving of current content
- Managed database entry and curation
- Supervised volunteers who will support the research, cataloging, and archiving initiative
- Maintained and updated SOPs relating to the archives
- Established procedures/guidelines for ingestion of new assets

ASSISTANT ARCHIVIST 11/2015–10/2016
UCL Petrie Museum of Egyptian Archaeology

- Provided reference guidance, class presentations, and public outreach
- Assisted Head Archivist with collection management
- Assisted with incoming research requests
- Assisted in preparing displays of collection materials, both in real and virtual exhibition spaces
- Digitally photographed and scanned archival materials to create high quality digital images

PROJECT ARCHAEOLOGIST/GIS SPECIALIST 10/2014–10/2015
UK Institute of Archaeology

- Prepared detailed maps for field projects and exhibitions
- Assisted with project planning and implementation
- Maintained central filing system and database for project planning documents and field maps

EDUCATION

Master of Arts Museum Studies, 2014 UK Institute of Archaeology 09/2011–05/2014	Master of Arts Egyptology, 2011 Cairo University Cairo, Egypt 09/2009–05/2011	B.A. Archaeology, 2009 University of Birmingham Birmingham, UK 09/2006–05/2009
---	--	---

Figure 21.1 A beautifully formatted example résumé

These kinds of résumé templates are very popular. They're often divided into two columns, often have little charts or "skill strength meters" like this one does, and may come complete with QR codes that link to a more expansive online résumé. For a human being reading this résumé, it's pretty eye-catching!

But the machines don't care. In fact, I've found that résumés like this—which are not dissimilar from what LinkedIn will export from your profile, if you ask it to—get mangled pretty badly when a machine tries to read it.

You should really have two résumés for each job. The first and primary one is the one that a human *can* read, but that a machine can *definitely* read. In other words, it'll be a bit plain, but it'll contain everything. Your secondary résumé can have amazing formatting and be eye-catching; it's the one you'll hand to people in person when you meet them.

Some key formatting tips for that primary résumé:

- Always use a word processing document, and almost always expect that to be Microsoft Word (.DOC or .DOCX format). Never submit a PDF to a machine if you can help it, because the machine stands a much higher chance of not parsing the text correctly from a PDF.
- Go heavier on bullet lists, and make sure your keyword checklist is well-represented. Keywords should be used organically in sentences. For example, *4+ years experience using Kubernetes in a DevOps environment and with CI/CD pipelines* is a great bullet that incorporates several keywords.
- I sometimes include a *very* short paragraph when I need to describe the "very subjective" items from the job posting. For example, along with the bullet points for a past job, I might write something like, "This job was especially exciting because it required me to be highly motivated and a self-starter. I was often handed projects and expected to find ways to complete those with minimal supervision, spending about 90% of my time working independently toward predefined outcomes." That's a way for me to pull in "very subjective" keywords like *highly motivated* and put them into the context of my work experience.
- No tables. No columns. No graphics. On the other hand, the machine won't care how long your résumé is, so don't limit yourself to one page.
- Do not use the header or footer features of your word processor; the machine often won't parse information from those sections. Put everything into the body of the résumé.

Finally, when you're finished, it's time to feed your résumé to a machine and see what happens. Numerous online résumé validators exist; search for "online résumé validator" and you'll probably find dozens. These "read" your résumé in much the same way a job website or corporate Candidate Management System would, letting you try out your résumé and see what the machine thinks. Some validators charge for the service while others require you to sign up for a mailing list, so use your discretion when choosing one. I actually send my résumé to a handful of different free

ones so I can get a variety of “machine opinions” on how well my résumé can be parsed.

21.3.7 Should you hire a résumé writer?

Professional résumé writing is a thing. It’s so much a thing, in fact, that there are professional certifications like the Certified Professional Résumé Writer (CPRW) that résumé writers can attain. I’m a professional writer. I’ve written almost a hundred books in total, and my readers often tell me that they can “hear your voice in my head when I read your stuff—you write just like you talk.” I take that as high praise, and it meant I felt pretty comfortable updating my résumé recently.

Still, I decided to spend a hundred bucks and have a professional résumé writer go over it. As a writer, one of my most valuable companions is my editor, and I figured a résumé writer could do something similar for me. Hoo-boy.

My skill at writing paragraphs of prose does not immediately translate to a good résumé, I discovered. I needed to be more concise, and had “Use bullet lists, Don!” beaten into my head pretty hard. I needed to use less “passionate technologist who aspires to lead a highly productive team focused on critical business outcomes” kind of language, and instead make more specific statements about my experiences and capabilities. (If you notice the flowery language on my LinkedIn profile, it’s because I often use it as a negative example in workshops.)

So, yes, hiring a résumé writer—especially for such a nominal amount—helped me. I’ve had friends who were helped by the résumé writer injecting a bit of human personality into their overly formal, robotic-sounding résumé. Others have spent more on what I’d call “résumé coaching,” to help whittle down a large and complex résumé into what would be the most effective for a particular job posting.

This isn’t to say that a résumé writer is a must-have. Many technology professionals get their next job through their network, which means their résumés are more a formality and less of a critical component to their job search (which is a huge argument for cultivating a professional network far in advance of needing it to help you land a job). But if you’re going to be relying on your résumé to get a job, it can sometimes be worth hiring a professional to help it read its best.

21.4 Nailing the interview

Interviews can be horrifically stressful for many of us. You’re dealing with a stranger—the interviewer—who has a part of your career in their hands. It’s a mystery what they’re looking for, and they’re often guided by prepared interview questions provided by their company’s hiring team—like the dreaded, “Tell me about your worst mistake at work” question. Here are some tips for a successful technology interview:

- Try to anticipate what questions you can, particularly ones that relate to items of experience requested by the job posting. For those questions, make sure you’ve prepared a concise answer of less than 30 seconds. Practice these answers ahead of time. For example, you should have answers about your experience with any

of the major technologies listed on the job posting, or any major skills—like leadership or communications—that the job posting highlighted.

- Always arrive with some concise, meaningful questions about the company itself. You’re about to become a partner with this company, and it’s good to know who you’re partnering with. Questions about the company vision, their management philosophy, how they feel about internal promotions and advancement, and similar questions reveal your interest in the company and can communicate an intent to make them a long-term partner.
- Pay close attention to body language. If someone looks bored, or leans back as if they’re not interested, wrap up whatever you’re saying and let them continue the interview.
- Bring concrete examples, if possible, of your work. That can include code samples (with previous employer information removed, of course), network diagrams, analytics dashboards, and other work products.
- Be prepared to talk about soft skills like communications, teamwork, leadership, and conflict resolution. Know your work style and be prepared to discuss it. These “soft” elements are often the core of the interview, where the interviewer is trying to see if you’ll “be a fit.” Help them arrive at that answer quickly and concisely.
- Watch your own body language. Fidgeting, failing to maintain eye contact, and other “tells” can make it more difficult for the interviewer to build trust and rapport. Practice with other people ahead of time. I’ve even gone so far as to head to a local bar and strike up conversations with willing strangers, just to help practice maintaining eye contact and not fidgeting (two things I struggle with).
- Think about how you reason through and solve problems—and prepare a concise explanation of that. Problem-solving skills are the most difficult thing to pull out of someone during an interview, and you can help the interviewer by doing that work for them, in advance.
- Never say “I don’t know” unless you follow with something like, “but here’s how I’d find out.” Technology professionals have to be confident self-led learners, and making that come out in an interview is a great bonus for you.

All of these tips are designed to make you come across as more prepared, more confident, and more able—so practice them!

21.5 **Understanding compensation packages**

I once worked for a company that did a great thing: every year, each employee would receive a custom statement of their compensation package. It included their salary of course, but it also included the value of things like

- Stock and options grants
- Health insurance and other paid benefits (which for that company included payments for a gym membership, college tuition reimbursement, and other benefits)

- Bonuses paid out
- Matching money paid to my retirement program

I thought that statement was great because it put a total financial figure on some things that a lot of employees don't even think about. Having fully-paid health benefits is, in the United States, a pretty standard item when you work for a tech company. Knowing that my company shelled out some \$10,000 on top of my salary for benefits was an eye-opener! That statement makes it easier to do an "apples to apples" comparison if you start looking at a new job. Will the new job cover my benefits also? Will there be reimbursement for wellness expenses, like a gym membership? Will the stock grants be as generous? How will the bonus program work? All of those elements of a compensation package are important to understand. With that in mind, I want to run through a few of the more common elements and explain them.

21.5.1 Compensation package elements

Keep in mind that some of these elements don't exist everywhere in the world, and some work very differently; I encourage you to use this list merely as an example, from which you can research and discover more about how things work in your part of the world:

- *Base salary*—This is an easy one, and it's the one we all think of first when it comes to compensation. Be aware that your employer often pays *more* than the amount shown. For example, in the United States, your employer pays your base salary to you. Deducted from your paycheck are federal income taxes, state income taxes (for states that have them), local income taxes (again, where applicable), and additional federal taxes for Social Security and Medicare programs. But your employer is also paying state unemployment insurance taxes and federal payroll taxes, which can amount to 10-12% of your salary. That's not deducted from your paycheck, but it does make you "more expensive" than your base salary implies.
- *Bonuses*—This is another common compensation element, and it differs drastically across organizations. Bonuses can take the form of profit-sharing programs, discretionary bonuses, bonuses based on personal and/or company performance, and many, many other things. Whatever program you're on, make sure you thoroughly understand how it works, and what factors impact what you receive. The promise of a 20% bonus sounds great—until you realize that it's based on unrealistic company revenue numbers that can't ever be achieved.
- *Stock grants*—A stock grant usually takes the form of a block of stock units, called *shares*, that *vest* to you over a period of time. For example, a grant for 10,000 shares might vest 12.5% of those shares every six months, meaning they would take 4 years to completely vest. The stock isn't actually *yours* until it vests, so this depends on you thinking you'll be with the company long enough for that to happen. Once it vests, you can generally *exercise* it, meaning you can sell it for cash at current market rates.

BEWARE OF TAXES! Make sure you ask a tax advisor about the tax implications of receiving and selling stock in your country. For example, in the United States, you can be forced to pay income tax on shares that vest to you. When you sell shares, you can pay additional income tax on any gain you realize, and that tax rate may depend on how long you've owned the shares.

- *Stock options*—An *option* is basically the right to purchase a share in the future, for a predetermined price. For example, suppose you were granted 1,000 options with a *strike price* of \$20. The options might vest over the course of 4 years. When a block of options does vest, you would have to make a choice: do nothing, or spend \$20 per option to purchase the associated shares of stock. The idea is that the stock will go up in price, meaning you might spend \$20 to buy a \$40 share, giving you a net gain of \$40 if you were to immediately sell that share on the market.
- *Insurance*—Many companies will offer free insurance of various kinds, and might offer the ability to buy discounted insurance as well, with the fees held from your paychecks. I've seen companies offer life insurance, pet insurance, legal insurance, disability insurance, and any number of other offerings. These are often discounted, sometimes significantly so, from what you'd pay to buy these on your own, but they're only valid so long as you remain at that employer.
- *Health care*—In countries (like the United States) without a broad public health-care program, most citizens get their health insurance from their employer. Some employers pay the entire amount, and some may even pay for your entire family. Others pay a partial amount, and deduct the rest from your paycheck. Some may pay nothing, and only offer a plan that you pay the full amount for. Many employers will also offer a variety of plans, with different coverage options and price points.
- *Retirement*—In the United States, it's common for employers to offer a 401(k) retirement program. This is essentially an investment account, and you're permitted to contribute up to a certain amount each year, taken from your paycheck before taxes are withheld. The amount in the program can be invested in stocks and bonds, enabling the amount to grow over time. Some employers will "match" a certain part of your contribution (how much varies widely between employers), boosting your savings.
- *Miscellaneous benefits*—These can range from reimbursing gym memberships or college tuition fees, to on-site lunch cafes with reduced prices, to free snacks in the break rooms, to a variety of other "perks."

There's no "right answer" on a compensation package: what's important to you and your family will determine what's "right" for you. But understand that companies look at the *entire* package when they're negotiating a salary for you. You might want \$10,000 more, but they're looking at the \$20,000 they may end up putting into your 401(k)

and thinking, “No, we don’t want to go any higher, because the other benefits make up for it.”

Also be aware that, in most jurisdictions, companies are required to provide certain benefits to all employees. For example, if a company says they’ll match 50% of your 401(k) contribution, they have to offer that for every employee. You can’t ask them to not offer that to you in exchange for more base salary; they’re not permitted to do that.

21.5.2 Negotiating your compensation package

With all of those basic compensation elements defined, how can you negotiate the best package for yourself when applying for a job? Start by asking how the company calculates compensation. A few companies these days use a strict *market compensation* process, whereby they establish pay ranges for each role, based on what the market in general is paying for that role. Within that range, a more experienced person will be paid closer to the top, while a less experienced person is paid closer to the bottom. With those companies, your ability to negotiate is often limited to your mutual assessment of your experience, and you should focus your negotiation efforts in that direction. Other companies may take a more subjective approach to compensation, which potentially widens your room to negotiate, but also makes it really hard to guess what a good number is.

Start by making sure you understand the *entire* compensation package, and the true value of it. Some employers offer extremely generous packages, and may throw in “lifestyle” elements—like the ability to work from home, or a promise to limit the amount of business travel you’ll be required to do—that make a particular base salary less critical.

Once you understand the entire package, *do not lowball yourself*. Make sure you know the salary range for your position in your field in your area. To find out this information, do some up-front research, using sites like Glassdoor.com or by performing internet searches for terms like “senior front end web developer salary tulsa, ok.” Remember that salaries do vary by geography, so make sure you’re looking at data that’s appropriate for where you live. Other places for “salary intelligence” include

- [Payscale.com](#)
- [Salary.com](#)
- [Indeed.com](#)
- Salary reports issued by various tech job search firms (which you’ll have to search for)

Most job applicants will try to ask for a little bit more than they think they’ll get, and that’s a good negotiating tactic since it leaves the employer room to negotiate you down. So long as you have a data-based expectation of what “reasonable” is, it’s fine to ask for 5-10% more and see what happens. Don’t accept less than that “reasonable” amount unless you understand specifically why (which means being comfortable asking that question of your prospective employer).

KEEP YOUR PREVIOUS OR CURRENT SALARY OFF THE TABLE Some employers will ask you about your previous salaries, or current salary, in an attempt to use that as a starting point for your new salary. I personally regard that tactic as distasteful, and it is illegal in some jurisdictions—do some research to discover if it's legal where you live. I advise answering that question with, "My previous compensation is confidential, and I don't see it having any bearing on our current conversation."

You should typically have a data-based salary number in mind when you go into the negotiation, and you should feel free to state that number up front. Make it clear that you're happy to communicate your market value and how that justifies your salary (and make sure you're prepared to do so, based on your research).

21.6 Further reading

- *Ladders Resume Guide*, Marc Cenedella (Ladders Press, 2019)
- *Mastering a Winning Resume: The Ultimate Guide to Beat the ATS, Impress the Recruiter, and Land the Interview Fast!*, Dan Reed (independently published, 2019)
- "8 Salary Negotiation Tips from Recruiters in Tech," Deborah Tennen (<https://zapier.com/blog/how-to-negotiate-salary>)

21.7 Action items

For this chapter, I'd like you to embrace the "it's never too early to start the job hunt" ethos, and complete a few tasks to get your job-hunt toolbox started:

- Head over to LinkedIn and spend some time thoroughly updating your profile. If you're not a LinkedIn user, glance at what a typical LinkedIn profile looks like (mine is at <https://linkedin.com/in/concentrateddon> if you want to use that one), and construct something similar in a word processing document. Commit to updating this quarterly.
- As a practice, conduct a pre-job-hunt review of your brand, particularly your social media footprint. Perhaps look at it with a colleague, focusing on the aspects that could be accessed by someone you don't know. Is there anything there you wouldn't want a potential employer to see? If so, what can you do to remedy that?

index

A

abstract concepts 185
accrual basis 197
achievements 222–223
active voice 117
agree (in RAPID framework) 164
alternatives, as lever 99
ambiguous language 118
analogies 183
appearance, physical 25–26
architecture, business 203–207
arguing 136
arithmetic average 200
arrogance, confidence vs. 110
assertiveness 126–127
ATSS (Applicant Tracking Systems) 213
audience 12–15
authenticity 210
autocorrect 111
averages 200–201

B

base salary 228
benefits 229
billing 197
blaming others 92
blogging 25, 33, 116
Bloomfield, Matt 31
Blue Sky mode 54–55
body language 26
bonuses 228
brand 12–20
 action items 20
 audience 12–15

contributing to success definition 20
further reading 19
professionalism 18–19
reach of 17–18
reviewing 210–213
 feedback 212–213
 professional brand 210
 public footprint 211–212
sabotaging 19
social media and 15–17
what is says 20
Bruner, Jerome 183
bullet points 214
business 147–162
 action items 162
 architecture 203–207
 further reading 162
 making money 153–155
 math and terminology for 195–208
 action items 207–208
 averages 200–201
 calculating self value 195–197
 further reading 207
 OpEx and CapEx 201–203
 P&L (profit & loss) statements 197–201
relationships 147–152
 changes in 152
 customers and employees 149
 one-sided 149–151
 overview 148–149
risk and reward 159–162
sold, recognizing what is 156–159
 details of business 159
 examples 156–159
business cards 27

C

calendar 73
 capabilities 203
 CapEx (capital expenses) 201–203
 driving business decisions 202–203
 overview 201–202
 career 3–11
 action items 11
 creating career plans 9–11
 job, success, and self 3–4
 life definition 4–7
 success 7–9
 technology communities value to 30–32
 career objective, stating 213
 career-safe skills 38
 Carey, Benedict 180
 cash basis 197
 cash burn 202
 cash on hand 189–190
 causation vs. correlation 175
 cautions or warnings, when to bring up 185
 Certified Professional Résumé Writer
 (CPRW) 226
 character 56
 Cirillo, Francesco 61
 COGS (Cost of Goods Sold) 156, 199
 commitment, group 77
 communications
 verbal 120–129
 action items 128–129
 assertiveness 126–127
 defeaters 123–126
 fear of speaking 121–122
 persuasion and art of listening 127–128
 respectful 77
 stepping up to 120–121
 written 104–119
 action items 118–119
 defeaters 117–118
 fear of communicating 108–112
 practicing 116
 storytelling 104–108, 113–116
 community work 222–223
 compensation packages 227–231
 elements of 228–230
 negotiating 230–231
 compromise 167
 conferences 23, 34
 confidence 110
 confident networkers 28–29
 conflicts 130–137
 action items 137
 decision-making frameworks 136
 deliberate 131

relying on data 135–136
 resolving 77
 returning to first principles 134–135
 seeking context 131–134
 winning vs. outcome 137
 connection 115
 constructionism 183
 constructivism 183
 contexts
 in conflicts 131–134
 leadership and 88–91
 contributing to technology communities 32–34
 conversations, in-person networking 26–27
 core competencies 221
 corporation, origin of term 147
 correlation vs. causation 174
 Covey, Stephen 166
 CPRW (Certified Professional Résumé
 Writer) 226
 credit 189–190
 critical thinking 139–144
 customers 149
 CV (curriculum vitae) 213

D

daily
 building muscles for 44–45
 tips for 46–47
 danger-zone skills 38
 data
 beware of 145–146
 relying on 135–136
 database administrator 43
 data-driven thinkers 138–146
 action items 146
 beware of data 145–146
 critical thinking 139–144
 decision-making 173–175
 further reading 146
 overview 144–145
 decide (in RAPID framework) 164
 decision-making 163–177
 action items 177
 being data-driven 173–175
 frameworks 136, 163–166
 further reading 177
 models 166–169
 Objectives and Key Results (OKRs) 167–168
 priorities 168–169
 rocks and pebbles 166–167
 negotiating 175–177
 OpEx and CapEx 202–203
 opportunity cost 169–170
 providing options 170–173

defeaters
verbal 123–126
practice staying on track 125–126
using voice as instrument 123–125
verbal fillers 123
written 117–118
ambiguous and pretentious language
118
passive voice 117–118
deliberate conflicts 131
deliverables, as lever 99
depreciation 202
details 51–53
digital communications problems 22–23
dirty work 78
disability insurance 192
discipline 58–59
dishonesty 92
Driscoll, Adam 30
drop-shipping 157

E

education 179
81 Fresh & Fun Critical-Thinking Activities
(Rozakis) 146
email 116
empathy 106
employees 149
entities, businesses as 147
entitlement 150
etiquette
for networking 25–28
in person 25–27
on LinkedIn 27–28
for technology communities 34–35
in open source projects 35
on Q&A websites 35
events, conference-style 23
expenses 199–200

F

factual statements 139
failure, supporting 77
family experiences 157
family, remote work and 70–71
fear
of communicating 108–112
addressing causes of 110–111
analyzing causes of 109–110
conquering 111–112
of speaking 121–122
being judged 122
not having all answers 121–122

feedback 212–213
fillers, verbal 123
filtering 51
first principles 134–135
following others 78
Ford, Neal 41
formatting résumés 224–226
frameworks, decision-making 136,
163–166
fresh tech skills 37–39
fully loaded salaries 196
functional problems 151
functions 203
furniture 69

G

GAAP (Generally Acceptable Accounting Practices) 199
genuineness 210
goals, preparedness 189
grammar 111
gross profit margin 199

H

hashtags on social media 25
health care 229
Helmick, Jason 30
help, asking for 77–78
helping others 178–187
action items 187
analogies 183
further reading 187
hands-on involvement 182
how humans learn 180–181
qualifying to help 179–180
toxic relationships 179
worthy of 179–180
reasons for 178–179
repetition 181–182
rest time 186
sequencing 185–186
Socratic method 183–185
Hicks, Jeffery 30
hiring résumé writers 226
How to Lie with Statistics (Huff) 201
How We Learn (Carey) 180
Huff, Darrell 201

I

icebreakers, in-person networking 26–27
ICYMI (In Case You Missed It) roundup 34
ideas, stifling 92

inclusive workplaces 80–82
 helping 80–81
 offering respect and support
 81–82
 income 197
 incompetence, level of 94–95
 in-person networking 25–27
 body language 26
 business cards 27
 icebreakers/ start conversation
 26–27
 physical appearance 25–26
 input (in RAPID framework) 164
 instant messages 116
 insurance 191–192
 interviews 226–227

J

Januszko, Missy 30
 JIT (just-in-time) stocking 156
 job hunt tools 192–193, 209–231
 action items 231
 compensation packages 227–231
 elements of 228–230
 negotiating 230–231
 further reading 231
 interviews 226–227
 résumés 213–226
 analyzing job postings 216–220
 being unique 215
 formatting 224–226
 hiring writers 226
 rules 213–214
 starting 214–215
 writing 220–223
 reviewing brand 210–213
 feedback from others 212–213
 professional brand 210
 public footprint 211–212
 tasks to do now 209–210
 job postings 216–220
 hard requirements 217
 keywords 219–220
 preferences 218–219
 job, career vs. 3–4
 job-safe skills 38
 just-in-time (JIT) stocking 156

K

Kanakos, Mike 30
 Key Results 168
 KRs (Key Results) 167

L

language
 ambiguous and pretentious 118
 passive voice 117–118
 laziness 58–59
 leadership 84–96
 action items 96
 further reading 96
 getting into different contexts
 88–91
 management vs. 85–86
 mistakes 92
 positive 91–92
 reasons for not moving into 94–96
 learning leadership 95
 level of incompetence 94–95
 measuring success 95–96
 taking care of team 93–94
 vision of 86–88
 learning 180
 about diverse topics 47
 daily
 building muscles for 44–45
 tips for 46–47
 how humans learn 180–181
 leadership 95
 media 45
 scheduling 47
 leasing 202
 LeMaire, Chrissy 30
 level of incompetence 94–95
 levers 99–100
 life definition 4
 life insurance 192
 LinkedIn 27–28
 listening 78, 127–128
 local user groups 23
 loss of income 189

M

management, leadership vs. 85–86
 market compensation 230
 market valuation 155
 Marquette, Kevin 31
 math 195–208
 action items 207–208
 averages 200–201
 business architecture 203–207
 calculating self value 195–197
 further reading 207
 OpEx and CapEx 201–203
 driving business decisions 202–203
 overview 201–202

math (*continued*)

- P&L (profit & loss) statements 197–201
 - expenses 199–200
 - revenue 197–199

Maurer, Doug 31

Medd, Jonathan 31

media, learning 45

medical insurance 192

memories 180

mentors 95

miscellaneous benefits 229

mistakes 93

mode, in sample set 201

models, decision-making 166

money

- as lever 99

- businesses making 153–155

Month of Lunches book series (Manning Publications) 186

mortgage insurance 192

multitasking 63–64

N

names, personal 28

negotiating

- decision-making 175–177

- solutions 100–102

network engineers 43

networking 21–29

- action items 29

- confident networker, becoming 28–29

- digital communications problems 22–23

- etiquette for 25–28

- in person 25–27

- on LinkedIn 27–28

- in person 23

- online 24–25

- reasons for 21–22

- remote work 73

neurons 180

note-taking 121

O

Obama, Barak 123

objective data 136

objective preferences 218

objectives 168

OKRs (Objectives and Key Results) 168

one-sided relationships 149–151

online networking 24–25

OOP (object-oriented programming) 183

open source books 34

open source projects 24, 34–35

OpEx (operational expenses)

- driving business decisions 202–203

- overview 201–202

opinion, use of term 139

opportunity cost 154, 169–170

options, offering 171

ordinary income 198

outcomes

- focusing on 223

- winning vs. 137

owners of businesses, risk and 160

P

P&L (profit & loss) statements 197–201

- expenses 199–200

- revenue 197–199

passive voice 117–118

perception 158

perform (in RAPID framework) 164

permanent remote work 74

personal life, sharing 77

persuasion 127–128

Peter Principle, The (Peter) 94

Peter, Laurence J. 94

Petty, James 30

physical appearance 25–26

Platinum Rule 79

podcasts 34

pomodoro technique 61–63

positive leadership 91–92

positiveness 78

postings, job 216–220

- hard requirements 217

- keywords 219–220

- preferences 218–219

practicing communication 116, 125–126

preparedness 188–194

- action items 193–194

- cash on hand and credit 189–190

- goals 189

- insurance 191–192

- job hunt tools 192–193

- possible problems 188

- social safety nets 190–191

pretentious language 118

priorities 168–169

problem-solving 97–103

- action items 102–103

- clearly stating problem 97–98

- identifying levers 99–100

- negotiating solutions 100–102

- troubleshooting vs. 97

procrastination 58–59

professional brand 178, 210

professionalism 49–57
 action items 57
 Blue Sky mode 54–55
 brand and 18–19
 drawing yellow lines 55–57
 focusing on details 51–53
 quit, knowing when to 53–54
 rules for 49–51
 being easy to work with 50–51
 delivering on promises 50
 saying no 49–50
 proficient tech skills 40–43
 promises
 delivering on 50
 saying no 49–50
 proof, to back up statements 140
 properties 183
 public footprint 211–212

Q

Q&A (question-and-answers) websites 24, 35
 qualifications summary 220–221
 questions
 anticipating 122
 asking 28, 171
 quitting 53–54

R

RAPID (Recommend, Agree, Perform, Input, Decide) 164
 rationing time 61–63
 reach, brand 17–18
 recommend (in RAPID framework) 164
 references, offering 214
 relationships 147–152
 as lever 99
 changes in 152
 customers and employees 149
 one-sided 149–151
 overview 148–149
 toxic 179
 relevant tech skills 40–44
 assessing 46
 focus of learning 43–44
 fresh vs. 37–39
 proficient 40–43
 reliable sources 141
 remote work 66–74
 action items 74
 adopting routine 71–72
 challenges of 66–67

creating space
 overview 67–68
 suggestions for 68–70
 explicitly defining culture 72–73
 networking 73
 permanent or temporary 74
 working with family 70–71
 renting space 69
 repetition 181–182
 respect 81–82
 rest time 186
 résumés 213–226
 being unique 215
 formatting 224–226
 hiring writers 226
 job postings 216–220
 hard requirements 217
 keywords 219–220
 preferences 218–219
 rules 213–214
 starting 214–215
 writing 220–223
 community work and achievements 222–223
 core competencies 221
 focusing on outcomes 223
 qualifications summary 220–221
 work experience 221–222
 retirement 229
 revenue 197–199
 reward 159–162
 Richards, Mark 41
 Ring, Jan Egil 30
 risk 159–162
 Robbins, Mike 31
 rocks and pebbles model 166–167
 role models 179
 routine 71–72
 Rozakis, Laurie 146
 rules
 professionalism 49–51
 being easy to work with 50–51
 delivering on promises 50
 saying no 49–50
 résumés 213–214
 storytelling 105

S

sabotaging brand 19
 self
 calculating value 195–197
 career vs. 3–4
 selling 156–159
 details of business 159
 examples 156–159

sequencing 185–186
services 203
Seven Habits of Highly-Effective People, The (Covey) 166
shares 228
Shell of an Idea: the Untold History of PowerShell (Jones) 21
skills 39, 46
social media 15–17
social safety nets 190–191
Socratic method 183–185
software developer 43
solutions 100–102, 114
space, for remote work
 overview 67–68
 suggestions for 68–70
stock grants 228
stock options 229
storytelling 104–108
 applying structure to 113–116
 applying to business communication 106–107
 everyday communication 107–108
 rules of 105
study groups 47
subjective preferences 218
subscription income 199
subscription service 156
success 7–9
 career vs. 3–4
 measuring 95–96
supervisors 85
support, inclusive workplaces 81–82
survival 181
synaptic networks 180
systems administrator 43

T

talking too much 93
teaching 178
teamwork 75–83
 action items 83
 advantages and disadvantages of 75–76
 being better team player 76–79
 contributing to inclusive workplace 80–82
 helping 80–81
 offering respect and support 81–82
 dealing with less-effective teammates 79–80
 further reading 82–83
tech skills 37–48
 action items 47–48
 daily learning
 building muscles for 44–45
 tips for 46–47

further reading 47
learning media 45
relevant 40–44
assessing 46
focus of learning 43–44
fresh vs. 37–39
proficient 40–43
tech-lead positions 95
technology communities 30–36
 action items 36
 contributing to 32–34
 etiquette for participating 34–35
 in open source projects 35
 on Q&A websites 35
 value to career 30–32
temporary remote work 74
term life insurance 192
theories 139–140
time management 58–65
 action items 65
 discipline, procrastination, and laziness 58–59
 multitasking 63–64
 pomodoro technique 61–63
 time catalog 63
 TimeFlip technique 59–61
tools, understanding 78
toxic relationships 179
translation 33
troubleshooting, problem-solving vs. 97

U

unbiased sources 141
user groups 33

V

verbal communications 120–129
 action items 128–129
 assertiveness 126–127
 defeaters 123–126
 practice staying on track 125–126
 using voice as instrument 123–125
 verbal fillers 123
 fear of speaking 121–122
 being judged 122
 not having all answers 121–122
 persuasion and art of listening 127–128
 stepping up to 120–121
videos 33
vision, leadership 86–88
voice
 as instrument 123–125
 passive 117–118

W

- Weltner, Tobias 30
whole life insurance 192
winning, outcome vs. 137
work experience 221–222
written communications 104–119
 action items 118–119
 defeaters 117–118
 ambiguous and pretentious language 118
 passive voice 117–118
 fear of 108–112
 addressing causes of 110–111
 analyzing causes of 109–110
 conquering 111–112
 practicing 116

résumés 220–223

- community work and achievements 222–223
 - core competencies 221
 - focusing on outcomes 223
 - qualifications summary 220–221
 - work experience 221–222
- storytelling 104–108
- applying structure to 113–116
 - applying to business communication 106–107
 - everyday communication 107–108
 - rules of 105

Y

-
- yellow lines 55–57

Own Your Tech Career

Soft skills for technologists

Don Jones

A successful technology career demands more than just technical ability. Achieving your goals requires clear communication, top-notch time management, and a knack for navigating business needs. Master the “soft skills,” and you’ll have a smoother path to success and satisfaction, however you define that for yourself.

Own Your Tech Career: Soft skills for technologists helps you get what you want out of your technology career. You’ll start by defining your ambition—whether that’s a salary, a job title, a flexible schedule, or something else. Once you know where you’re going, this book’s adaptable advice guides your journey. You’ll learn conflict resolution and teamwork, master nine rules of professionalism, and build the confidence and skill you need to stay on the path you’ve set for yourself.

WHAT'S INSIDE

- Personal branding and career maintenance
- Barriers that derail progress
- The rules of the business world
- Market analysis to keep tech skills fresh

For tech professionals who want to take control of their career.

Microsoft MVP **Don Jones** brings his years of experience as a successful IT trainer to this engaging guide.

Register this print book to get free access to all ebook formats.
Visit <https://www.manning.com/freebook>



"Valuable advice for every stage of your tech career."

Lee M. Cottrell
Pittsburgh Technical College

"A must-have book for those interested in pursuing a successful and satisfying IT career."

Irfan Ullah
Shaheed Benazir Bhutto University

"A unique book on how to build a successful tech career and lead a satisfying life!"

Ed Lo, Uber Technologies

"I recommend the book to everyone!."

Markus Braasch, Credit Karma