

Table of Contents

Preface	xv
1. Programmer to Engineer	1
An Engineer by Any Other Name	1
Fundamentals Matter	2
The Many Paths to Becoming a Software Engineer	3
What You Were Taught Versus What You Need to Know	4
Embrace the Lazy Programmer Ethos	5
The Value of a Fresh Set of Eyes	7
Don't Solution Too Quickly	8
Apply the Golden Rule to Software	11
Wrapping Up	11
Putting It into Practice	12
Additional Resources	12
2. Reading Code	15
The Challenge of Working with Existing Code	15
Cognitive Biases	17
Approaching Unfamiliar Code	18
Software Archeology	20
Effective Code-Reading Strategies	25
Leveraging IDE Features	25
Reading Tests for Insight	31
Practice Makes Perfect	33
Wrapping Up	34
Putting It into Practice	35
Additional Resources	35

3. Writing Code.....	37
Don't Reinvent the Wheel	38
What Is Good Code?	39
Less Is More	41
The Zeroth Law of Computer Science	42
Beware Boilerplate Code	42
Favor Composition over Inheritance	42
Favor Short Methods	43
Write Code to Be Read	43
The Problem with Code Comments	45
Tests as Documentation	46
Avoid Clever Code	47
Code Reviews	48
Avoid the Checkbox Code Review	48
It Is Hard to Be Criticized	49
Fostering Trust	50
Learning New Languages	50
Wrapping Up	52
Putting It into Practice	52
Additional Resources	53
4. Modeling.....	55
What Is Software Modeling and Why Do We Do It?	56
Which Diagrams Do You Need?	58
Context Diagrams	60
Component Diagrams	60
Class Diagrams	61
Sequence Diagrams	62
Deployment Diagrams	63
Data Models	65
Additional Diagrams	66
Modeling Best Practices	67
Keep It Simple	67
Know Your Audience	68
Be Careful with Your Color Choices	68
Establish Standards and Templates	69
Tools	69
Wrapping Up	71
Putting It into Practice	71
Additional Resources	72

5. Automated Testing.....	73
Benefits of Automated Testing	73
Acts as Documentation	74
Improves Maintainability	74
Boosts Your Confidence	75
Leads to Consistency and Repeatability	76
Types of Automated Testing	77
Unit Tests	78
Integration Tests	78
End-to-End Tests	79
What Mix of Tests Should You Be Writing?	79
What You Should Not Test	80
Code Coverage	80
Writing Tests	81
Getting Started	81
Assertions	82
Writing Unit Tests	83
Mocking	84
Writing Integration Tests	86
Writing End-to-End Tests	87
Wrapping Up	88
Putting It into Practice	88
Additional Resources	90
6. Exploring and Modifying Unfamiliar Systems.....	91
Understanding Unfamiliar Codebases	91
Start with the Big Picture	92
Understand the Execution Flow	95
Build Mental Models Incrementally	105
A Sample Process	109
Making Changes Safely	110
Refactoring Safely	111
The Scout Rule	112
Small, Reversible Changes	114
Wrapping Up	118
Putting It into Practice	118
Additional Resources	119
7. User Interface Design.....	121
Designing for Everyone	121
What Is Usability?	123
What Is Accessibility?	124

What Are Localization and Internationalization?.....	125
Know Your User	126
Secondary Users	127
You Are Not Your User	127
Impact of Culture	129
Maximizing Usability	130
Principles of Design	131
Contrast	131
Repetition	132
Alignment	132
Proximity	133
Applying the Principles of Design	135
Make the Right Thing the Obvious Thing	139
The Importance of Good Error Messages	145
Destructive Actions	146
Wrapping Up	149
Putting It into Practice	149
Additional Resources	150
8. Working with Data.....	151
Understanding Data Types and Formats	152
Structured Versus Unstructured Data	152
Common Data Formats	155
Specialized Data Considerations	158
Storing Your Data Effectively.....	160
Database Types and Their Use Cases	161
Data Persistence and Management	166
Database Connections and Transactions	170
Consistency Models and Caching Strategies	171
Planning for Data Growth	177
Querying and Managing Data Performance	179
Efficient Query Writing	179
Tools and Best Practices	183
Data Migration and Transformation	188
Understanding Data Movement Fundamentals	188
Handling Schema Changes	190
Wrapping Up	193
Putting It into Practice	194
Additional Resources	195
9. Software Architecture.....	197
What Is Architecture?	197

Trade-Offs	199
Architecture Versus Design	201
Quality Attributes	202
Identifying Quality Attributes	203
Gaining Stakeholder Alignment	204
Architectural Styles	206
The Agile Architect	206
Fitness Functions	208
Architectural Diagrams	209
Architectural Decision Records	210
Wrapping Up	212
Putting It into Practice	212
Additional Resources	213
10. To Production	215
The Complexities of Production Environments	216
Users Are Unpredictable	216
“But It Works on My Machine”	217
Building Production-Ready Code	220
Performance Optimization	220
Environment-Specific Configurations	221
Error Handling and Logging	225
Security Essentials	226
Deployment Pipeline	233
Deployment Environments	234
Version Control Strategies	235
Deployment Automation	239
Deployment Strategies	243
Continuous Integration and Continuous Deployment	244
Production System Monitoring and Maintenance	248
Monitoring	248
System Maintenance	249
Wrapping Up	251
Putting It into Practice	252
Additional Resources	253
11. Powering Up Your Productivity	255
Optimizing Your Development Environment	255
Know Your Development Tools	256
Build Your Own Lightsaber	256
Leverage the Power of the Command Line	258
Harness the Power of Keyboard Shortcuts	260

Strategic Automation	263
The Perpetual Pursuit of Productive Habits	264
Collaborative Learning	264
Personal Knowledge Management	265
Wrapping It Up	267
Putting It into Practice	267
Additional Resources	268
12. Learning to Learn.....	269
Cramming Doesn't Work	270
Skills Acquisition	271
The Learning Habit	273
Learning Through AI	275
Fear of Missing Out	278
Where Should You Invest Your Time?	280
Practice Innovation	281
Architectural Briefings	283
Practice Grace	284
Wrapping Up	285
Putting It into Practice	285
Additional Resources	286
13. Mastering Soft Skills in the Tech World.....	287
Collaborative Communication	288
Communication Channels	288
Enterprise Operator	295
Know Your Audience	297
Practicing Influence	297
Understanding and Articulating Value	298
Strategic Approaches to Influence	298
Stakeholder Management	300
Time Management	301
Maker's Schedule	302
Staying on Task	303
Wrapping Up	304
Putting It into Practice	304
Additional Resources	304
14. Career Management.....	305
Plan Your Career Path	305
Find What You're Passionate About	306
Exploring Your Career Options	308

Walking Backward from Your Goals	312
Deliberate Skill Acquisition	313
Aligning Career Choices with Life Phases	316
Walking Your Career Path	319
Celebrate and Record Your Wins	319
Overcome Imposter Syndrome	320
Build Your Professional Community	322
Cultivating Your Professional Relationships	325
Acing Your Next Interview	326
Create Work–Life Balance	334
Wrapping Up	338
Putting It into Practice	339
Additional Resources	340
15. The AI-Powered Software Engineer.....	341
What Is AI Really?	342
Demystifying AI Terminology	342
Understanding AI’s Capabilities and Limitations	348
AI as Your Pair Programmer	351
Standalone Chatbot Assistants	351
Inline IDE Assistants	352
Agentic AI IDE Environments	353
Prompt Engineering Fundamentals	355
What Is Prompt Engineering?	355
Essential Prompt Engineering Techniques	356
Advanced Prompt Engineering Techniques	358
How AI Might Shape Software Engineering	360
Will AI Take My Job?	361
Vibe Code Reviews	362
AI as Your Force Multiplier: From Writing Code to Problem-Solving	365
Wrapping Up	367
Putting It into Practice	368
Additional Resources	368
Index.....	371