

PRINCIPLES
FOR
PROGRAMMERS

THE CONDENSED ADVICE FROM THE WORLD'S BEST

BY ANDREI NEAGOIE

v1.0.0

Subscribe to get updates and new principles as this book evolves at zerotomastery.io

Copyright © 2020 by Andrei Neagoie

All rights reserved. This book or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the publisher except for the use of brief quotations in a book review.

PREFACE

Advice is easy to give. That is why advice that is given should always be inspected, debated, and analyzed. What may have worked for one individual will not necessarily work for everyone or in all situations.

At the same time, as humans, we are able to learn from the past and stand on the shoulders of giants. This way, we don't make the same mistakes over and over. Instead, we get to learn from the mistakes of the past so that we start on our journey along a path that isn't completely unknown. If we are able to jump start from other's experiences, and be efficient in our learning, that is a winning strategy that will take us far.

This is the reason for this book. For the past 5 years, I have been collecting advice, articles, books and opinions of some of the best programmers, thought leaders, and business people that I know. I wanted to collect ideas from people with different opinions, and narrow

down these recommendations to the core and essence. Sometimes I include principles coming from my own experience, but the goal of this book is to combine not just my own advice, but to combine the advice of the top performers and find the commonalities between them so that you don't have to start from the very beginning.

Whether you are just learning to code, or you have many years of experience in the industry, you should find something actionable and valuable in this book for your own situation.

This book is not meant to be read in one sitting. Each principle will offer its own value. Think of these as tidbits that you can read every once in a while, come back to, or be reminded of something important, so that you make sure you are on the correct path to success.

Enjoy, and let's take the leap.

- *Andrei Neagoie* ([@andreineagioe](https://twitter.com/andreineagioe))



WHAT'S NOT GOING TO CHANGE IN THE NEXT 10 YEARS?

Build a strategy around the things that are stable in time. As a programmer, you will always have to solve problems. You will always have to effectively communicate with others, and you will always have to continue to learn and upgrade your skills. Focus a big chunk of your energy on skills that are virtually guaranteed to be valuable in the future, no matter what.



It's impossible to imagine a future 10 years from now where a customer comes up and says, "Jeff, I love Amazon; I just wish the prices were a little higher." "I love Amazon; I just wish you'd deliver a little more slowly." Impossible. When you have something that you know is true, even over the long term, you can afford to put a lot of energy into it.

— JEFF BEZOS



HELPING OTHERS WITHOUT ANSWERS

Programmers don't grow by being given the answers. They grow by trying things, experimenting and solving problems themselves. These are the skills and lessons that will serve them well as they level up their career. Instead of giving them the answers, give them the tools to find the answers. Help others by letting them reach a conclusion or solution on their own. Just guide them in getting to that goal without holding their hand.





BEWARE OF THE HALF-LIFE

Everything we know, not just about programming, has an expiration. The half-life of knowledge is a term to denote the time it takes for half the knowledge in a domain to be outdated. According to one study, the half life of an engineer in 1930 was 35 years vs 10 years in 1960¹. Imagine what it is now. Because technology is moving so fast, eventually the ability to solve problems, and continuously learn, will become the most sought after skill. This is a muscle that you can train.



1. <https://spectrum.ieee.org/riskfactor/computing/it/an-engineering-career-only-a-young-persons-game>



EXPERIENCE IS NOT CREATED EQUAL

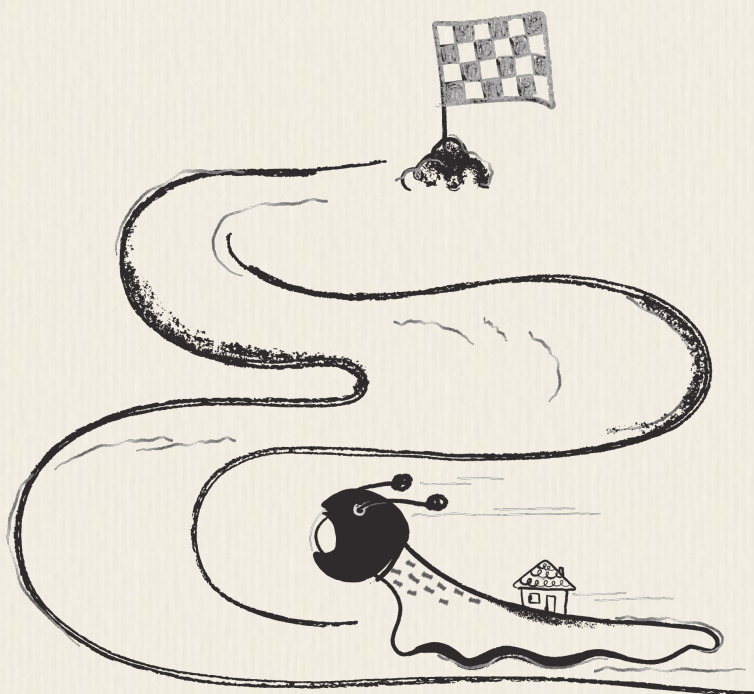
10 years coding by yourself with no other programmers around you, with no code reviews, doesn't necessarily make you better than someone who has 1 year of coding experience working on a project with team members and getting code reviews daily. Get out of your bubble. Work with others, be surrounded by other programmers and learn from them. The biggest thing you can do if you are starting out is to get hired as soon as possible so you can work with others to accelerate your learning. Let others see your work. Get feedback. That's how you gain experience.





DON'T FOLLOW THE MONEY, FOLLOW THE BRAINS

Great pay and salary today is good. But if you want to succeed long term, you are better off taking a lower pay to be surrounded by smart people and being in a situation that allows you to learn from others. If you can get both, that is even better, but always have this as a priority when picking between options: Which option will allow me to learn and grow the most and be surrounded by people smarter than me?



I hope you enjoyed the first five principles.

Get the full ebook on Amazon at:

[LINKS.ZEROTOMASTERY.IO/EBOOK](https://links.zerotomastery.io/ebook)

Find out more about my work and the classes I teach at:

[WWW.ZEROTOMASTERY.IO](https://www.zerotomastery.io)