

# How do you get more work done in less time?

---

 <https://codewithoutrules.com/secretskills/>

Itamar Turner-Trauring

Fri Aug 13 00:36

## Tired of scrambling to get your work done?

There's always more work to do.

There's more code to write, more features to add, more bugs to fix. There are always more issues queued up in the issue tracker, more maintenance that you just can't get to, more ideas coming from the product manager.

There's always more work, but there's never enough time. Even if you worked 24 hours a day, 7 days a week, you would never be able to keep up with the backlog on your software project.

But if you were productive enough, you could:

- Take the afternoon off, confident that you've produced highly valuable work.
- Gain more influence on how your project or team runs.
- Have an easier time finding a new job – including jobs with a technology stack you don't know.

How do you become more productive as a programmer? **By creating more value with the fixed time you have.**

This might seem a like truism, so let's dig a little deeper and see where the value you produce really comes from.

## Moving beyond churning out code

Let's say you've been asked to code a specific implementation of a specific feature: it should take you five business days to implement. If you get distracted, spin your wheels, or notice a bad assumption too late, you might end up taking two weeks instead of one: you've cut the amount of value you produced in half.

So obviously, you want to avoid wasting time while you're coding.

**But what if you came up with a better and faster implementation strategy, that took only one day to code?** You'd be producing the same amount of value in one day instead of five: you'd be operating at  $5 \times$  the productivity.

Coming up with good solutions is what senior software engineers do, and that's a big improvement in productivity over just coding what you're told to code. But even here you're still dependent on someone else to come up with valuable problems to solve: if you're given a low-value problem, your productivity will be low no matter how good your solution is.

When you take the next step and start identifying problems, when you can go to your boss and say "I've noticed this bug that's losing users to the tune of \$3 million a year, shall I go fix it?" , *that's* when you're really providing value. **The key to productivity is learning how to identify and solve problems—before you ever start coding.**

## How do you learn these skills?

Unfortunately, most discussions of programming productivity end up focusing purely on technology and skip over these necessary problem-solving skills. That's why they're "secret" : unless you have a really good manager you're expected to somehow figure them out on your own.

**Knowing these problem-solving skills can be the difference between working overtime and getting a raise, between taking months to find a new job and quickly getting a job you love.**

I've learned these skills over the course of 20 years, but you can learn them far faster by reading my new book, *The Secret Skills of Productive Programmers*.

The book covers over 30 different skills and techniques, including:

- **Efficient implementation skills** so you don't waste time: timeboxing, how and when to ask for help, which order you should implement subtasks, and more.
- **Finding better solutions** using techniques like the 3-solution rule, research, negotiation, designing in writing, and more.
- **Identifying more valuable problems** using techniques like decision critiques, joint search with solutions, and more.

## Get the book

As of early October 2020, the book isn't *quite* done, but it is converging on its final form: the content is there, and I've done a first editing pass. Remaining work includes additional editing, and making sure the Kindle and other book reader files are working correctly. At that point I will be raising the price.

Instead of waiting for that last 10% of polishing and editing, you can **buy the book now for less money, get productive faster, and get all future updates to this edition (delivered via email).**

You have two options for getting the book:

## Option #1: Donate to support Black lives

Across the US, people are protesting ongoing police violence and brutality towards Black people, and in particular the ongoing murders of Black people by police. One thing you can do to help support the protests is donate money:

1. Donate \$20 (or \$50, or \$200) either to a local BlackLivesMatter group, the [Movement for Black Lives Fund](#) (more about the M4BL [here](#)), or to [a local or national bail fund](#) to help get protesters out of jail.
2. Forward your receipt (make sure there's no credit card digits) to [donations@codewithoutrules.com](mailto:donations@codewithoutrules.com).

I will then email you a free copy of the book. As is the case when you buy the book, you will also get future updates to the 1st edition (via email).

**Over the past few months, readers donated over \$5000.** Why not join them in taking a small, easy step to make the world a better place?

## Option #2: Buy the book

Alternatively, you can just buy the book—unless you were offended by the above, in which case I'd prefer not to take your money.

### Buy the book

US\$19 + taxes

- Money back guarantee, for any reason.
- 160-page book in PDF format, and alpha versions for Kindle (mobi) and other ebook reader (epub).
- All future updates to the 1st edition (delivered via email).
- The fast path to learning the productivity skills you need.

[Buy now \(\\$19\)](#)

## Changelog

### September 22, 2020

Added extensive example of designing in writing.

### August 10, 2020

Initial versions of mobi and epub files, still very preliminary.

Complete editing pass on the whole book:

- Adjusted chapter order.
- Typo and phrasing fixes throughout.
- Rewrote part of a number of chapters.

## June 24, 2020

- Added new chapters on defining the problem and learning the zeitgeist.
- Added section on searching for errors to search engine chapter.

## June 1, 2020

- Greatly expanded the chapter on learning from others.
- Added an interview with Alex Gaynor.

## May 18, 2020

- Expanded the chapter on close reading with an example of close reading of documentation.
- New chapters: how to use a search engine better, systems not individuals.

## May 8, 2020

- New chapters: Why not The Five Whys, Walking away, Choosing which problem to work on.
- Moved close reading chapter to solutions section.

## May 6, 2020

Initial release.

## About the author



Hi! I'm Itamar. I've been writing software since 1995 or so. As an employee I've worked for companies small and big, ranging from 8-person startups to a year at Google as a product manager (my previous employer got acquired). And as a consultant I've written software for a similarly broad range of companies.

I wrote this book in the hopes of helping you and other programmers learn the productivity skills that took me decades to learn.

Want to learn more about me? I write a weekly newsletter sharing my past programming and career mistakes, and how [you can avoid them](#).