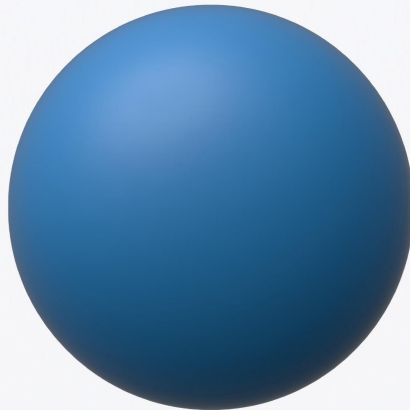


PragmaticBookshelf

# simplicity

sustainable, humane, and  
effective software development



dave thomas

edited by  
Susannah Davidson

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit <https://www.pragprog.com>.

Copyright © The Pragmatic Programmers, LLC.

# Let's Change Our World—Again

---

Software development is inherently complex.

Unfortunately, we seem to delight in making it not just complex, but complicated, too.

What's the difference? A snowflake is complex; the weather system that produced it is complicated. Both have many parts, but with the snowflake you can see how they interact; there's a pattern, and some rules.

With software, complexity is difficult, but it has rules. Complications arise when we don't follow them. And these complications turn our projects into hard-to-understand, hard-to-predict, tangled balls of logic. They interfere with the way we run our projects, and the way we deal with people.

Working with complications is like playing a game where you not only don't know the rules, but where you can't know the rules.

The sad thing? Most of the complications in our projects are self-induced. We add them when we don't stop to think, when we don't take the time to listen to that small inner voice saying "really?".

We don't invest the time to make things simple, because we have no time. We have no time because things are complicated.

We have to make the time; if we don't, we won't learn and improve. We have to discover how to embrace simplicity and create simple things, otherwise we'll spend our time like the plate juggler, running between the poles to stop things from crashing down.

Ward Cunningham coined the phrase "do the simplest thing that could possibly work." That's not just a way of writing code; it applies to everything we do.

Simplicity doesn't mean simplistic. It doesn't mean naïve. It means producing things that are easy to understand and change, and that have that vague quality of "feeling right." Christopher Alexander, the architect who wrote the

original [A Pattern Language: Towns, Buildings, Construction \[AIS77\]](#), called this the *Quality Without A Name*. Achieving simplicity rewards you with this feeling.

## Don't Do What I Do

There's a problem writing about simplicity; there are no rules or procedures that make things simple, no coding standards or design methodologies that guarantee the things you produce will not end up complicated. Simplicity isn't the way you do things; it's the spirit with which you do them.

Why? Because what's seems simple to me might not seem so to you, and something you find simple might go over my head.

I cannot tell you what to do, because I don't know. Instead, I describe situations that I've been in that felt more complex than they should be, and then outline the steps I took to simplify things. I don't want you to adopt any practice unthinkingly. Instead, I hope that you'll read each as a kind of story: the path I took to make something simpler. Treat this as an example as you find your own path.

## Courage

Kent Beck kicked off modern software development with [Extreme Programming Explained: Embrace Change \[Bec00\]](#). He lists the five values of XP as Communication, Feedback, Respect, Simplicity, and Courage. I believe these are core values not just for eXtreme Programming, not just for programming in general, but for life.

Ultimately the ideas in this book all make great demands on you as a person. They ask you to create and follow a set of values, often in the face of peer pressure. They ask you to think about what you're doing, rather than going with the flow, and they inform you when you have to make a stand. And they ask you to fight against all the factors that make things complex as you try to bring simplicity into your life.

All this takes courage. All great changes do.

Enjoy the process.

## See You Online

<https://pragprog.com/titles/dtcode>

The book's home page, with links to its discussion forum and the errata list

<https://articles.pragdave.me>

Where I write articles and notes about simplicity and whatever else catches my fancy.

<https://pragdave.me>

My personal homepage.

<https://pragprog.com>

My business and the last 20 years of my life.

## My Thanks

To my reviewers: Aino Corry, Bruce Eckel, Derek Sivers, Kim Shrier, Neal Ford, Noel Rappin, Saron Yitbarek, and Zachary McKenna. True friends are the people who tell you you're wrong without making you feel like a complete fool.

To my editor: This is the first time in 25 years of writing books that I've had an editor, and I'm never going back. Susannah Davidson is incredibly skilled at seeing the big picture and the small details, all at the same time, and finding ways to make them coherent. Another friend.

To my wife: for being my wife *and* my friend.

And, finally, to all of you, for listening.

Part I

Simplify What You Do And How You Do  
It

