

March 2025

Avoid the nightmare bicycle

In my opinion, one of the most important ideas in product design is to avoid the “nightmare bicycle”.

Imagine a bicycle where the product manager said: “people don’t get math so we can’t have numbered gears. We need labeled buttons for gravel mode, downhill mode, ...”

This is the hypothetical “nightmare bicycle” that Andrea diSessa imagines in his book [Changing Minds](#).

As he points out: it would be terrible! We’d lose the intuitive understanding of how to use the gears to solve any situation we encounter. Which mode do you use for gravel + downhill?

It turns out, anyone can understand numbered gears totally fine after a bit of practice. People are capable!

Along the same lines: one of the worst misconceptions in product design is that a microwave needs to have a button for every thing you could possibly cook: “popcorn”, “chicken”, “potato”, “frozen vegetable”, bla bla bla.

You really don’t! You can just have a time (and power) button. People will figure out how to cook stuff.

Good designs expose systematic structure; they lean on their users’ ability to understand this structure and apply it to new situations. We were born for this.

Bad designs paper over the structure with superficial labels that hide the underlying system, inhibiting their users’ ability to actually build a clear model in their heads.

computational medium must be comprehensible in its own terms. This section is about what "comprehensible in its own terms" might mean. Comprehensibility is the main topic of this chapter.

When I was about twelve years old, I had a recurrent nightmare. My old balloon-tired, second-hand bicycle had about run its course as a prized possession. I was beginning to lust after a fancy new three-speed English racing bike. The nightmare was that I got my dream bike, with a twist. Instead of an elegant 1, 2, 3 on the gear shifter, my new bike had a long series of icons showing the use to which each gear was suited: smooth pavement uphill, smooth pavement downhill, gravel, crossing railroad tracks, and others I had no way to decipher (although inscrutable icons is not the point of this example).

It is a little difficult to understand how these symbols could strike terror in the heart of a twelve-year-old boy, but my memory of my alarm is vivid. In any case, the nightmare makes a good foil to wring out some of the design considerations and aesthetics that explain why Boxer is the way it is. Although that nightmare now seems to me more bizarre than frightening, I still find many modern systems truly horrifying when they are designed according to principles like the ones used for that strange bike.

What could the designer of the nightmare bike have been thinking? It is actually not difficult to see. He (let's say) conceived of the device he was designing as accomplishing a number of tasks for its operator. He thought: one needs to go uphill or downhill, travel on paved and gravelly roads, cross railroad tracks, and so on. To this observation we add one classic maxim of design, "make form follow function," which we learn from television advertisements is the practice of the best automobile designers. The design of the gear-changing mechanism follows directly: for each function, there is a corresponding structure, a "gear" that suits that function.

It's worthwhile going through what is wrong with this design to illustrate what turn out to be quite central points about comprehensible and useful design. First, my nightmare gear shift is a mess. How would I ever know where to find the gravel position, except by memorization? How would I know if there is or should be a gear for uphill, gravelly railroad tracks? There is no sense of systematicity or completeness in the

mechanism. Does this sound a little like some annoyances you've found in computer applications?

Our hypothetical designer has ignored some powerful order and simplicity that might be achieved by focusing a bit more on structure, rather than on function. Order the gears in a mathematical sequence based on increasing rate of revolution of the bike wheels per stroke of the pedal: 1, 2, 3, . . . I can hear the executive talking to the engineer who came up with such an "orderly" design at Nightmare Bikes, Inc.: "Who do you think we're designing for—an engineer? That's math; that's ratios. Customers want a job done, like going uphill or down. They want form to follow function."

Of course, the executive is mistaken, as we know from our experience with bicycles, but I can guarantee you that the scenario is being played out constantly in appliance and software companies with consequences at least as bizarre as the nightmare bike. Our home hand mixer is labeled stir, fold, blend, puddings, mix, cakes, mash, cream, frosting, beat, whip, and egg whites. It's all function, with no hint of orderly structure.

How do people manage to understand bicycle gearing if, as is no secret, ratios and gearing are not a particular strength of the general populace? Their success is not "just experience," although that is not to be ignored. Instead, learning bicycle gearing is another example of one of the grand, overarching learning principles we introduced early in the book and elaborated in chapters 4 and 5. There are great resources in people's rich store of intuitive knowledge, resources that are vastly underappreciated and underused. In this case, people know about sequence and patterns of "more or less" earlier than they learn to count. Babies pile and collect things; they have early senses of more and less and how to achieve them. Before they can talk, they know "do that *again*; I want *more*." Possibly imperfect but powerful ideas like these, plus a few pieces of experience, make bicycle gears comprehensible. Mathematical p-prims from everyone's early experience with the world can substitute, sometimes, for a formal understanding of ratios.

Think about some learning experiences. How long will it take a young biker to realize that one direction of gear changing is more generally apt for going faster and the other direction for going slower? You don't need numbers or any concept of gearing to appreciate that. Or, "I need more

Two parts
of a job

depend on how
"instructive" it
is

"reducing noise"

making
use cases

Systematic
product

p.s. Changing Minds is one of the best books ever written about design and computational thinking, you should go read it.

Subscribe

I periodically write about programming tools, end-user programming, and other software topics. To get updates about new posts:

[Join my email newsletter](#)
[Follow me on Twitter](#)
[Subscribe via RSS](#)