

## **Handling Design Criticism**

Rebecca Wirfs-Brock

Vol. 24, No. 3  
May/June 2007

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.



© 2007 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

For more information, please see [www.ieee.org/portal/pages/about/documentation/copyright/polilink.html](http://www.ieee.org/portal/pages/about/documentation/copyright/polilink.html).

# Handling Design Criticism

**Rebecca J. Wirfs-Brock**

*The art of being wise is the art of knowing what to overlook. —William James*

**D**esign reviews are an essential part of any design process. However, taking the criticism that comes from such reviews can be hard. The word criticism even has a slightly negative connotation in our culture. But design criticism is invaluable, and effectively giving and receiving it are skills that every software designer needs to master.



It can be difficult to filter out constructive arguments from the noise or to discern the reasoning behind offhand remarks. Knowing what tactic to take when someone criticizes your design can keep your creative design juices flowing and help you improve your ideas. Here's a summary of some kinds of criticism

you might receive and how you might react.

## **Valid criticism**

Suppose someone points out a flaw or design weakness that you perceive as valid. Your best response might not be to give up your idea, especially if you're presenting a novel approach. Maybe your idea just needs refinement. You might ask your critic what improvements she suggests. But those who criticize don't always have ready fixes for your design flaws—instead, they might propose an alternative. Design alternatives need to be weighed and considered, too.

Don't overreact to criticism. Don't immediately throw out your design in the face of criticism. Think before discarding your design in favor of a hastily proposed alternative. You

don't have to handle criticism in real time. The most frustrating yet satisfying design sessions in my career occurred when we left problems unresolved only to come back the next day with fresh perspectives and work through the rough bits with ease. That agonizing end-of-day pause forced our brains to continue working on the problem in the background.

## **Judgmental criticism**

If someone remarks "That won't work" or "I don't like your solution" or "Your design stinks!" he's being judgmental. There might be valid reasons for his judgment, but you won't know what they are unless you press for clarification. To get beyond judgments, you need specifics. But sometimes, when pressed, people give goofy reasons. Nigel Warburton's slim volume, *Thinking from A to Z* (Routledge, 2000), neatly defines and illustrates many illogical arguments. When I spot an illogical argument in a discussion and put a name to it, it helps me think about how I might alter the discussion or shed light on that argument's irrelevancy. Effectively counteracting false arguments can require finesse. Appeals to reason don't always work. But at least I know what I'm up against. Let me mention a few illogical arguments I find especially relevant.

A false dichotomy is when someone sets up a situation so that only two conclusions seem possible. If someone proposes an alternative to your design using this strategy, she will paint her solution as the good one and yours as the poorer alternative. But there might be other possibilities. Both solutions might be stinkers! There might be a raft of other options worth

exploring. You might need time to form a reasoned opinion about the counterproposal. The best way to counteract a false dichotomy is to point it out and offer a more systematic consideration of alternatives.

Even more insidious is calling for a vote on a design decision—especially in haste. A democratic fallacy treats majority opinion as revealed by voting as a source of truth and a reliable guide for action. In many cases, taking a vote is an extremely unreliable way of discovering the most appropriate choice, especially if the majority is largely ignorant or indifferent of various design options.

Another favorite of mine is the slippery-slope argument: “Well, if we do as you suggest, what will prevent us from applying that solution here, and there, and over there, with dire consequences?” The implication is that once you start down a path, you’ll be forced to continue on that path to an ultimate (bad) conclusion. I don’t like solutions that appear to bend a design out of shape or seem like compromises. Perhaps that’s why I find slippery-slope arguments so compelling when considering one-off solutions. Valuing consistency and clarity leads me to repeat a choice once it’s established as credible.

But slippery-slope arguments obscure the fact that rational human beings can decide how far down a slope to traverse. Just claiming there’s a slippery slope shouldn’t be enough to squelch an idea. To judge a design fairly, I need to honestly assess the alleged inevitability of the supposed slide. People daily make rational decisions to take exception to established convention. If you make a performance optimization here, you don’t have to make it there. If a proposed design breaks a preestablished pattern, it’s good to question it. Although creatures of habit, we can make reasoned choices. And good solutions don’t always fit with the status quo.

### Invalid criticism

If someone makes a comment that’s clearly off base, he might have a different perspective on your design. If you want to pursue his reasoning, you need to figure out why he holds his view. He



might be basing his remarks on bad experiences. He might hold some unspoken bias. Or he simply might not understand your reasoning.

If lack of understanding is causing him to make an invalid criticism, the burden is on you, the designer, to communicate your design intent and rationale. Maybe you’ve obscured your design with too many facts. I once advised a designer who got a lot of flak about her framework to present two different views—one targeted at users and another, more detailed presentation for those who would extend it. Once she explained those different perspectives, the criticism evaporated.

Critics might need to see different aspects before they fully understand. Perhaps concrete examples, a picture

or rough sketch, some code, or some tests to demonstrate your software’s behavior will help. Maybe you need to offer some rationale in addition to your technical discussion. You need to address your critics’ concerns if you want to sway their opinion.

### Personal criticism

Instead of making a judgment about your idea, the criticizer levels a judgment about you: “You’re stupid!” Ouch. Someone who attacks your integrity isn’t playing fair. Although easier said than done, the best course of action is to not rise to the bait.

### Aesthetic criticism

Someone comments that she doesn’t believe your solution fits her ideal of

“good.” There are many acceptable ways to solve a design problem; she dislikes yours. Every designer has personal design aesthetics and needs the freedom to approach a design problem in his or her own way. If you don’t have this freedom, you might become a mere note taker for a committee, and design by committee seldom produces high-quality designs. Ultimately, someone has to own a design and make the aesthetic decisions (you’re fortunate if you and your colleagues share identical aesthetic values).

It’s okay for critics to point out aesthetic issues, and a designer might actually learn something from these criticisms. But, by definition, aesthetic issues are never critical to a design. Don’t get hung up on them.

Critics don’t always realize that their criticism is about aesthetics. They might think it’s more fundamental. As a designer, you need to clarify this with your critic and try to find mutual understanding about whether the issue is technical or aesthetic. If you’re responsible for solving that part, then you should be able to exert some personal choices. That is, unless your solution sticks out like a contrarian sore thumb. Inconsistency is often a valid critique. But persistent aesthetic arguments are usually a sign of clashing values.

I remember being asked to reconcile differences between two experts who were clashing over the design of a graphics class library. I was the neutral object expert who could exert “good design values” and help them reach a reasoned compromise. But they were both strong willed and unbending. I was able to get them to listen to each other but couldn’t get them to agree on anything. The project stalled and eventually was cancelled. Strongly held aesthetic views are rarely swayed by logic or reason. However, I know of someone who recently defused an aesthetic disagreement by asking the designer and his critic to work out their ideas in more detail. When they reconvened, the one who had initially leveled the criticism conceded that the other’s solution was reasonable. This expert defused the situation by calling for a dispassionate exploration of ideas.

## Complexity criticism

Someone complains that your proposed solution is overly complex. “You aren’t gonna need it” is the mantra of many agile developers. The YAGNI catchphrase reminds developers not to overengineer solutions. But this slogan, when casually tossed into a review, can be inflammatory. The idea behind that catchphrase is to not embellish your design; add necessary complexity at the last responsible moment. But what if you and your colleague disagree about when that moment might be? And sometimes, “too complex” arguments reflect a lack of understanding of the problem’s inherent complexity.

I recently participated in a discussion on building design consensus. An Extreme programmer recounted an experience where he argued for a more straightforward solution over the seemingly complex one that his programming partner offered. His partner didn’t believe his own solution *was* overly complex. Complexity is in the eye of the beholder (and those who hatch an idea might not be able to see it as others do). They had to agree to some strategy in order to move forward. So he proposed, “let’s not do it your way or my way; let’s try it my way,” which acknowledged the potential for reversing that decision. But to him, this was unsatisfying—one had to concede to the other.

While I favor simplicity, I don’t favor it at the expense of repetitive code. I find ways to counteract redundancy by making common behavior useful in more than one context. I might choose to increase complexity to accommodate variations. And at times I get pushback. If people propose simplifications, I’ll carefully listen to their arguments. I value simplicity, too.

## Compliments

Someone just said your design was good or great. Maybe it’s genuine praise. But if people pile on praise too often they might be cloaking their busyness or disinterest. Or perhaps they don’t feel comfortable leveling criticism because of cool reception to it in the past. There’s not much you can learn from a compliment unless you ferret out the underlying reasons behind the praise. I find it particularly difficult to probe for those reasons. I feel like I’m fishing for further compliments. But if I think I might get some real information, I offer thanks and ask, “What is it you particularly like about the way we did x?”

**A**s designers, we need to practice recognizing, accepting, and seeking valid criticism. Having others review our work and handling their accompanying criticism are essential to developing a good design. Sometimes you might face strong disagreement and harsh criticism. But design reviews aren’t a blanket invitation for reviewers to rework a design to reflect their own style and aesthetics. We designers need to learn how to quickly identify and filter the criticism that should be overlooked. A win-win solution is one where we don’t do it “your way” or “my way,” but a “reasonable way given what we know to be true at the moment.” If you can’t reach consensus on some point, you need to call for acceptance. That doesn’t mean your critics must favor your solution, just that they agree to support it. 🍷

**Rebecca J. Wirfs-Brock** is president of Wirfs-Brock Associates and an adjunct professor at Oregon Health & Science University. Contact her at [rebecca@wirfs-brock.com](mailto:rebecca@wirfs-brock.com).

## Correction

Owing to an editing error, the reference list in the Mar./Apr. Design column (“Driven to ... Discovering Your Design Values”) incorrectly listed Martin Fowler instead of Kent Beck as the author of *Test-Driven Development: By Example*. We apologize for this error.