

I usually call these endless discussions “religious debates,” because they have a lot in common with most discussions of religion and politics: They consist largely of people expressing strongly held personal beliefs about things that can’t be proven—supposedly in the interest of agreeing on the best way to do something important

(whether it's attaining eternal peace, governing effectively, or just designing Web pages). And, like most religious debates, they rarely result in anyone involved changing his or her point of view.

Besides wasting time, these arguments create tension and erode respect among team members, and can often prevent the team from making critical decisions.

Unfortunately, there are several forces at work in most Web teams that make these debates almost inevitable. In this chapter, I'll describe these forces, and explain what I think is the best antidote.

“Everybody likes _____.”

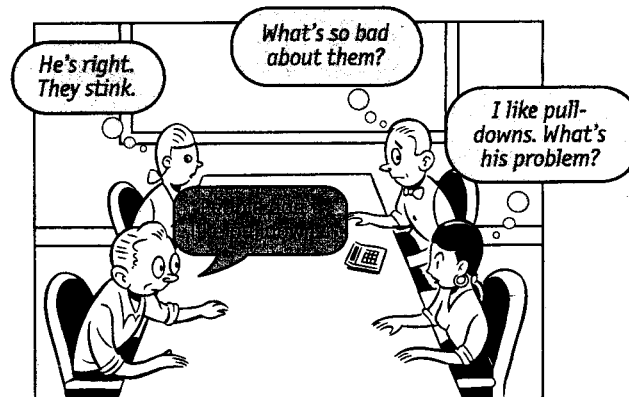
All of us who work on Web sites have one thing in common—we're also Web *users*. And like all Web users, we tend to have strong feelings about what we like and don't like about Web sites.

As individuals, we love Flash animations because they're cool; or we hate them because they take a long time to download. We love menus down the left side of each page because they're familiar and easy to use, or we hate them because they're so boring. We really enjoy using sites with _____, or we find _____ to be a royal pain.

And when we're working on a Web team, it turns out to be very hard to check those feelings at the door.

The result is usually a room full of individuals with strong personal convictions about what makes for a good Web site.

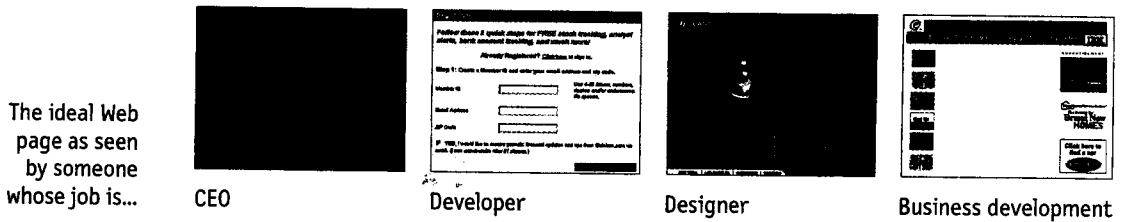
And given the strength of these convictions—and human nature—there's a natural tendency to project these likes and dislikes onto Web users in general: to think that most Web users like the same things we like. We tend to think that most Web users are like us.



It's not that we think that *everyone* is like us. We know there are *some* people out there who hate the things we love—after all, there are even some of them on our own Web team. But not *sensible* people. And there aren't many of them.

Farmers vs. cowmen

On top of this layer of personal passion, there's another layer: professional passion. Like the farmers and the cowmen in *Oklahoma!*, the players on a Web team have very different perspectives on what constitutes good Web design based on what they do for a living.¹



Take designers and developers, for instance. Designers tend to think that most people like sites that are visually interesting because *they* like sites that are visually interesting. In fact, they probably became designers because they enjoy good design; they find that it makes things more interesting and easier to understand.²

Developers, on the other hand, tend to think people like sites with lots of cool features because *they* like sites with lots of cool features.

The result is that designers want to build sites that look great, and developers want to build sites with interesting, original, elegant features. I'm not sure who's the farmer and who's the cowman in this picture, but I do know that their differences in perspective often lead to conflict—and hard feelings—when it comes time to establish design priorities.

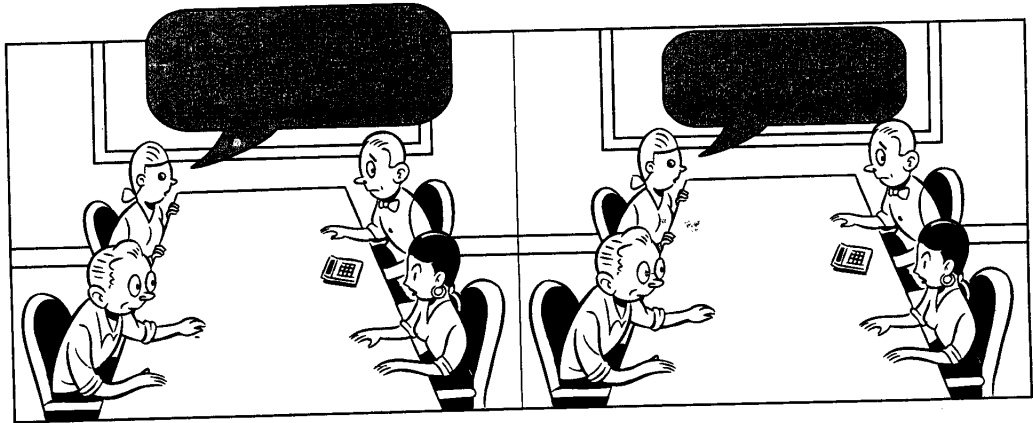
¹ In the play, the thrifty, God-fearing, family-oriented farmers are always at odds with the freewheeling, loose-living cowmen. Farmers love fences, cowmen love the open range.

² Yes, I'm dealing in stereotypes here. But I think they're useful stereotypes.

At the same time, designers and programmers find themselves siding together in another, larger clash between what Art Kleiner describes as the cultures of hype and craft.³

While the hype culture (upper management, marketing, and business development) is focused on making whatever promises are necessary to attract venture capital, users, strategic partners, and revenue-generating deals to the site, the burden of delivering on those promises lands on the shoulders of the craft culture artisans like the designers and programmers.

This Internet version of the perennial struggle between art and commerce (or perhaps farmers and cowmen vs. the railroad barons) adds another level of complexity to any discussions of usability issues—often in the form of apparently arbitrary edicts handed down from the hype side of the fence.⁴



³ See "Corporate Culture in Internet Time" in *strategy+business* magazine (www.strategy-business.com/culture_change/00103/).

⁴ I once saw a particularly puzzling feature on the Home page of a prominent—and otherwise sensibly designed—site. When I asked about it, I was told, "Oh, that. It came to our CEO in a dream, so we had to add it." True story.

The myth of the Average User

The belief that most Web users are like us is enough to produce gridlock in the average Web design meeting. But behind that belief lies another one, even more insidious: the belief that most Web users are like *anything*.

As soon as the clash of personal and professional opinions results in a stalemate, the conversation usually turns to finding some way (whether it's an expert opinion, research, focus groups, or user tests) to determine what *most* users like or don't like—to figure out what the Average Web User is really like. The only problem is, there is no Average User.

In fact, all of the time I've spent watching people use the Web has led me to the opposite conclusion: all Web users are unique, and all Web use is basically idiosyncratic.

The more you watch users carefully and listen to them articulate their intentions, motivations, and thought processes, the more you realize that their individual reactions to Web pages are based on so many variables that attempts to describe users in terms of one-dimensional likes and dislikes are futile and counter-productive. Good design, on the other hand, takes this complexity into account.

And the worst thing about the myth of the Average User is that it reinforces the idea that good Web design is largely a matter of figuring out what people like. It's an attractive notion: either pull-downs are good (because most people like them), or they're bad (because most people don't). You should have links to everything in the site on the Home page, or you shouldn't. Menus on the top work better than menus down the side. Frames, pages that scroll, etc. are either good or bad, black or white.

The problem is there *are* no simple "right" answers for most Web design questions (at least not for the important ones). What works is good, integrated design that fills a need—carefully thought out, well executed, and tested.

Take the use of Flash, for example.⁵ If asked, some percent of users will say they really like Flash, and an equal percent will probably say they hate it. But what they

⁵ *Flash*, Macromedia's tool for creating animated and interactive user interfaces, not *flash* (lowercase), the arbitrary use of whiz-bang features to make a site more interesting.

really hate is Flash used badly: large, complicated animations that take a long time to download and don't add any value. If you observe them carefully and ask the right questions, you'll likely find that these same people will appreciate sites that use small, hardworking, well-thought-out bits of Flash to add a pleasant bit of sizzle or useful functionality without getting in the way.

That's not to say that there aren't some things you should *never* do, and some things you should *rarely* do. There are some ways to design Web pages that are clearly wrong. It's just that they aren't the things that Web teams usually argue about.

The antidote for religious debates

The point is, it's not productive to ask questions like "Do most people like pulldown menus?" The right kind of question to ask is "Does *this* pulldown, with *these* items and *this* wording in *this* context on *this* page create a good experience for most people who are likely to use *this* site?"

And there's really only one way to answer that kind of question: testing. You have to use the collective skill, experience, creativity, and common sense of the team to build some version of the thing (even a crude version), then watch ordinary people carefully as they try to figure out what it is and how to use it.

There's no substitute for it.

Where debates about what people like waste time and drain the team's energy, testing tends to defuse arguments and break impasses by moving the discussion away from the realm of what's right or wrong and into the realm of what works or doesn't work. And by opening our eyes to just how varied users' motivations, perceptions, and responses are, testing makes it hard to keep thinking that all users are like us.

Can you tell that I think testing is a good thing?

The final three chapters are all about what, why, and how to test.