

Badass—An Interview with Kathy Sierra

BY NICKY BLEIEL | Senior Member

Kathy Sierra is the author of many successful books, including the award winning Head First series with Bert Bates. She's an in-demand conference speaker, trainer, programmer, and the founder of the online community JavaRanch. In this interview, she and Nicky Bleiel discuss her new book, Badass: Making Users Awesome.



Nicky Bleiel: Kathy, thanks so much for joining me to discuss your new book. I must admit, the minute I heard the title I wanted to interview you about it, because technical communicators consider our main goal “making users successful.” Your book ups that ante to “making users awesome.” That’s impressive.

Kathy Sierra: Thank you. That title was actually controversial. For a long time they didn’t want to use the word, “badass.” I chose it very deliberately, because I use to refer to it as making “passionate” users. But I realized how many time people mistook “passionate” to mean “people are really enthusiastic about the brand.” I always meant “growing expertise,” not “we love the brand” or “we love the product.”

I chose “badass” because it was just a more difficult word to get wrong.

NB: It is much less ambiguous. What was actually the impetus for writing *Badass*? It was many years in the making, correct?

KS: It was. I delivered it about six years late. I’m the worst possible role model for meeting a deadline but I had very patient publishers.

Badass started after we (Bert Bates and I) started the Head First series, with the first book, *Head First Java*. It wasn’t originally going to be a series. There was also a lot of controversy

about that book, because it didn’t look like a typical programming book. People who flipped through the pages, but didn’t actually study what it was doing, would say, “Oh, this is just really a dumb-downed book. It’s just basically cartoons.” They didn’t understand at all what was happening. They didn’t look at it. They didn’t realize that it actually went to a pretty advanced level.

When *Head First Java* became so successful—it became the number one programming book and 10 years later it is still the number one Java programming book. In fact, it is one of the longest running technical best sellers on Amazon. People always said, “Why? Why is this book successful?”

“Badass” started as a series of private talks for publishers and technical people, and a few very specific conferences. That’s how it really all evolved; it was our attempt to say, “This is why we did it this way.”

NB: To me, what separates this book from most books about building a better product or building a better user, is the fact that you used cognitive science to explain why your recommendations are sound. Could you explain a why you took this direction?

KS: Yes, cognitive science and other aspects of psychology. Cognitive science is familiar to me and to Bert,

because long ago we both worked in artificial intelligence. It just resonates for us. But really, the main thing is, I think it provides a way of “thinking about thinking” that’s broader and I think in many ways more useful than what has often been educational theory.

NB: In the book, you say that where we—the royal we—go wrong is that we tend to focus on our products when we should be focusing on our users. What exactly is a “badass” user and why are they better users?

KS: Yeah, you just asked the big question. There are several little pieces to this. The first one is that, when we buy a product, we’re not buying the product per se; we’re buying the thing that we intend to do with it. Some people refer to that as “the job that you’re hiring the product to do.” You have some context in which you want to use this product.

It’s very rarely about the thing itself. A simple example is one I used in the book: marketing for a high end camera is all about photography. It’s not about the camera. Then, as soon as you buy the camera, you’ve spent your \$2,000, you just get the manual—which is about nothing but the camera.

The goal when we buy a product—the goal when buying a camera—is not to just master the camera. It is photography.

Our job is to help people become better at the context in which they’re going to use a product. The extent to which we help people become better at the context is really the best help for creating the kinds of user experiences where people are likely to tell other people about that product.

Companies are focused on getting the user to love the company, love the brand. You need to switch your point of view to helping the user love what he is able to do, as a result of that product. Another way to say that is, people aren’t buying a product because they love the company, they’re buying the product—deep

down—because they love themselves. They want to do something that’s actually good for themselves.

A “badass” user is someone who is becoming more and more badass at the greater context, the context in which the product is used versus the product itself.

NB: In the book you take some jabs at common strategies for what people call “brand engagement,” such as gamification, social media likes, and viral videos. Why do you think these things can actually make users feel less engaged?

KS: There are a few reasons. I won’t go into all the details about the problems with gamification, because there’s a lot of work out there now that people can refer to. But what’s typically called “gamification,” is just operant conditioning, just another form of Skinner. It’s using the same psychological principles as gambling. That’s not exactly empowering to anyone. It’s certainly not producing more skills for the people who are “becoming engaged.”

But there’s another problem, which is gamification is literally consuming someone’s cognitive resources to engage in it. They cannot be simultaneously “engaging with your brand” and actually getting better at something at the same time, something they care about. We’re only shooting ourselves in the foot by trying to take their interest and attention when people have such a limited amount as it is.

NB: You note in your book that users need to get through what you call the “suck zone” before they could move onto “badass.” In a nutshell, what is the “suck zone” and how can we help users get through it faster?

KS: The “suck zone” is the part of the learning curve. Depending on how complex and difficult whatever it is that they’re learning—where it’s not yet in any way intrinsically rewarding and possibly feels really bad. It’s a

threshold. If you cross out of the “suck zone” then you at least have a chance to keep going. But we don’t want to lose people because they drop out while they’re still in that “suck zone.”

We also have to help them stay motivated while they’re in it. When I say motivated, I’m not referring to gamification-style motivation, but the kind of motivation that allows them to believe it’s worth it.

A lot of times we lose people, especially in difficult technical topics, because the topics are really difficult. They are challenging. I teach programming. There a lot of things that are challenging. We try to pretend like that’s not the case.

Part of the problem with a lot of technical documentation is not that the documentation itself is too difficult for people—it’s that we act as though it’s not. People drop out because they don’t realize that actively struggling is temporary and it’s typical. Sometimes all you have to do is say, “Yes, this is going to be difficult. This is part right here is going to be confusing. You’re actually not going to understand all that’s going on right now, but hang in there because it will all start to coalesce over time. Don’t worry about it, you can keep moving forward now.”

We know how easy it is to overwhelm short-term memory. We need to honor that. That’s part of keeping someone motivated.

NB: A little bit of truth telling could go a long way.

KS: Oh yeah, and very little.

NB: This book is appropriate for anyone who wants to create a successful product. It’s essentially a business book in a way, but there are many tips that can help technical communicators create better content.

I’d like to talk about those for a minute. You actually give a lot of great advice about what users need and why they need it. Can I throw out a few common deliverables and you share your opinions of them?

KS: Absolutely.

NB: “Tips and Tricks” documents.

KS: I think “Tips and Tricks” are great because tips and tricks can help provide that really intrinsically rewarding motivation for people when they’re still, for example, in the “suck zone” and as they move up the curve.

It can feel what it’s like to be making progress, even if they’re not yet there. I think “Tips and Tricks” are incredibly helpful. They also can be just a way of conserving people’s cognitive resources. If you can give them a tip or a trick, you’ve just spared their brain.

NB: Cheat Sheets, also known as Quick Reference Guides or Job Aids.

KS: Yes, and those are also fantastic for several reasons. Don Norman wrote *The Psychology of Everyday Things*. It’s very outdated but it doesn’t matter because the principles are so deeply useful. One of the things he popularized was this idea of “knowledge in the world” versus “knowledge in the head.” Any time you can place knowledge in the world you relieve the burden of that knowledge having to be in the person’s head.

NB: Quick Start Guides.

KS: A tutorial is what we call “teaching by remote control.” You’re being told which buttons to push and when to push them. You’re just moving through this thing. You’re not doing anything with enough repetition to actually be able to learn it, learn it deeply.

It’s not supported by research on memory. It’s not supported by research on perceptual learning. Having said that, it’s a great way for people to just get that overall feel of what the whole workflow is like.

NB: One thing that is very popular now is to create short videos that show people how to do a task. What do you think of those types of videos?

KS: The most useless but correct answer is, “Well, it depends.” I’m a big fan of video in a lot of different ways. Because showing you how to do it is almost always going to be better than having you read the words about it—and have to make the translation in your head to what that actually means—especially if there aren’t any graphics. In that case, it makes sense.

A lot of times, we put out these things, and we’re very fuzzy about what its intended use is supposed to be. Whether it’s to tell someone, “OK, you should really watch this about 12 times and practice it,” versus, “Oh, don’t worry about it. This is your cheat sheet. Just pull this video up, and just follow the steps.”

That’s always on us, I think, to give users that kind of guidance, because they don’t know.

NB: Jumping into user manuals themselves, you say in the book that user manuals can help users become badass. The problem is the manuals include everything “just in case” when our brains prefer “just in time.” Can you explain that, and how can we solve this problem?

KS: Yes. There are really a couple of issues, but they’re related. A lot of product manuals, they’re so specific to, again, the product. It’s like they’ve completely forgotten the context in which the user is using it.

Not including that context means that you’re giving the person’s brain a much harder task to pay attention to, because it doesn’t seem relevant. We create our materials as though that wasn’t even an issue.

“Just in case” learning is “here are all the facts you need to know about these things,” and “just in time” learning is “I need to learn this right now” or “I need to accomplish this task.” A lot of learning can’t be “just in time” learning. There are things you have to learn in advance. As I say in the book—I don’t want my neurosurgeon learning everything “just in time.”

If you wrap the things that you want them to learn “just in case” in

a contextual “just in time” example, then it helps their brain pay attention, and gives them something to hang this new content onto.

It helps their memory. It helps everything, it provides, usually, a much better experience.

So often, we’re looking at all these things and saying, “You’ve got to know this. You’ve got to know that.” We really have and make sure that all of those things are absolutely necessary. Are they necessary right now? We need to strip out all the edge cases, and all the “let’s learn every detail about X.” It’s taking all that stuff out, and pushing it into an appendix or a reference document, so that they can actually really nail something.

NB: That’s true and that takes us to something you mentioned in the book about optimizing content so that the “stuff that matters” gets in the user’s brain. That would be one of the ways that you would suggest doing that, correct?

KS: Yeah. It’s never made any sense to me. I mean how many software manuals have I read, including commercial books about those products—which is I think even more of a crime—is where they say, “Here’s the first chapter. Let’s look at the interface. Now, let’s look at all of the preferences. Let’s set up all the preferences.”

I (the reader) have no idea why those preferences actually matter. I don’t have the slightest idea. None of it is meaningful. You’re just overwhelming my cognitive resources instantly. Now, you’ve just made me incredibly less capable to actually go further and there’s no useful reason to do that.

Instead, we can put it in the documentation the set of trusted defaults and say, “Look, don’t worry about this now. You have all the control that you could ever want here.”

There’s a lot of research about what they call the “tyranny of choice.” Having to make decisions is very stressful because, first of all, we’re

often asking people to make decisions when they don’t even have all the information they need, or even any of the information they need. But just making a decision is now burning cognitive resources.

We need to be making those decisions for people. That doesn’t mean take away their ability to make the decision. What we take away is forcing them to make a decision or even making them think they need to make that decision right now.

NB: After reading the book, I get the feeling that you think, and you could correct me if I’m wrong, that a technical communicator’s overarching goal should be to act as a coach. Would I be reading that right?

KS: I think making sure that people always know that they will be able to do this, that they will be able to get through this. That it’s totally doable. Yes.

Again, at every point letting them know there will be a struggle, but they will get out the other side of this and it’s totally normal. You are right where you should be.

Because that helps keep people moving forward. Again, even if motivation isn’t your goal, which of course it should be, if we’re technical communicators, we always have that as a responsibility because if we don’t take motivation as our own responsibility—the motivation of the person actually learning, then we are actually potentially de-motivating them.

It really matters. We can’t say, “Well, it’s their job to be motivated. It’s my job just to provide the learning.” But that’s really not understanding how the brain works. I think it’s our job to pay attention to that motivation and keep them moving forward.

NB: Technical communicators consider themselves the “voice of the user.” In order to be the voice of the user, we have to know the user and understand the user, and also consider their context.

KS: Yes, we have to know the user. A lot of times, that's interpreted as "we have to know what level of knowledge they have coming in, or we have to know exactly what their job function is coming in."

We don't think about the fact that really "knowing the user" just means "I care who they are." We're forgetting is that they're actually humans. It's pretty easy to make anything better—without having any other knowledge about any specific people who are your users—by just saying, "They're humans. They have brains. Their cognitive resources are very limited."

NB: One of the key points you mentioned in the book is "if we really care about our users, we'll help them do what they want, not what we want." I think that has parallels to the technical communication mantra

"Give users what they need, when and where they need it." Do you see that also?

KS: Absolutely. That's really such a noble and useful and ethical goal. As you can imagine, there aren't a lot of groups in the company that often have that as their goal.

If you really look at cognitive resources as the brain's bank account, it's so limited every given day. It has nothing to do with intelligence. It has everything to do with what users have to actually think about, that they can spend those precious, scarce, easily depleted resources on.

Anything we can do to help support drain their resources less is... we're helping their whole life. We are quite literally giving them the opportunity to have more quality time with their kids at home. Those things

are so profoundly important, and the way we spend people's cognitive resources, we just waste them—either because we just don't give people anything that helps them or we give them things that deliberately actually make things worse for them.

It would also be giving them what they need, when they need it, and *not* to giving them things they don't need.

NB: I want to thank you so much for joining me, Kathy. I look forward to embracing my role as a coach and applying your advice about respecting cognitive resources in my work.

KS: Thank you so much. This was a really fabulous conversation for me. I know I'm having a conversation with someone who understands the jargon. It's just really great to be able to talk about it. **I**

As a member,
you get the tools
you need to
be successful.



- Education and Training
- Job Bank
- Award-Winning Publications
- Affinity Program
- Networking
- Chapters and Special Interest Groups
- Annual Conference
- Body of Knowledge
- Discounts and Special Offers



www.stc.org

ASD Simplified Technical English (STE) term checker

STE helps to make instructions as clear as possible. But, STE has many grammar rules and thousands of dictionary rules.

To remember all the rules is difficult. Can you remember the approval status of the word *fluid*?

- Not approved
- Approved as a noun
- Approved as an adjective
- Approved as a noun and as an adjective.

Can you remember your organization's technical terms and the unapproved alternatives?

The STE term checker from TechScribe helps you to make sure that your text conforms to STE.

Free trial: www.simplified-english.co.uk

* *Fluid* is approved as a noun only, unlike *liquid*, which is approved as a noun and as an adjective.

Techscribe®