Bringing Design to Software
Addison-Wesley, 1996

8

The Designer's Stance

An interview with David Kelley by Bradley Hartfield

The designer has a passion for doing something that fits somebody's needs, but that is not just a simple fix. The designer has a dream that goes beyond what exists, rather than fixing what exists

David Kelley is one of the most visible product designers in the world, especially in the world of high technology. He founded and heads IDEO, a firm that has been widely recognized for quality in design (Profile 8), and teaches product design at Stanford University. Although most of the products that Kelley and his firm have developed over the years are not primarily computer software, the firm has been one of the pioneers in interaction design.

In this interview with Bradley Hartfield, who is also a software designer and teacher in human–computer interaction, Kelley explores the boundary between design and engineering, and describes the place for creativity and openness in design. He emphasizes the degree to which successful design depends on a design mindset: one that is open to possibilities and ready to take risks in a *creative leap* into possibilities that are not yet defined and whose consequences are not yet visible. Kelley's apparent message is not surprising: Good design takes creativity. His deeper message is that creativity is not a matter of genius or of innate talent. A designer can learn and can be trained to develop openness and a willingness to take the creative leap. The designer's environment also plays a major role—the flow of interactions with other people in that environment can either encourage or stifle creative openness. The challenge to the software-design profession is to foster environments, both in teaching and in industry, that enhance the designer's potential.

— Terry Winograd

§ § §

Bradley Hartfield: David, let's start with a central issue. What is distinctive about design and designers versus other constructive fields and their practitioners?

David Kelley: There is a basic difference between problem solving and actually creating beyond what the problem calls for. It might help to pose two caricatures—two hypothetical extremes. One is engineering as problem solving; the other is design as creating. In this view, engineers are problem solvers. "This device breaks down when you use it for a long time, so we'll beef up the strength of this weak section"—that's problem solving. The designer, on the other hand, has a passion for doing something that fits somebody's needs, but that is not just a simple fix. The designer has a dream that goes beyond what exists, rather than fixing what exists.

You can fix a problem with a creative solution, of course, but I'm saying that that's not *all* the designer wants to do—the designer wants to create a solution that fits in a deeper situational or social sense. It is interesting that people with design experience are able to see the nuances of what makes something appropriate in design. They don't have any trouble coming up with views; they are critical about products. They think beyond the obvious.

BH: Let's explore the differences between engineering and design. My sense is that engineers are taught a particular way of looking at the world—a particular way of assessing situations—and are given a particular set of tools that they can use to address those situations. They are highly trained in mathematics, so they think that they can have a strong degree of certainty—of proof and absolute truth. From my own experience and from talking to other designers, my sense is that design is almost the opposite of mathematics or engineering. Certainty just is not one of the features that you have—none of your tools can lead to it, none of your insights yield it.

DK: I like to say that design is messy. Engineering—as it is taught, if not in practice—is not supposed to be messy. In engineering, you try to assume away the messiness in formulating the problem. You adopt a starting assumption, such as "Assume a spherical cow." That works only when you are solving a well-formulated problem in which, for the relevant practical purposes, cows can be treated as spheres**.**

I sometimes give my engineering students problems that begin "Assume that this is a linear situation," which is absolutely false. All the interesting design problems show up when stuff is not linear. But, engineering works with a set of rules, so you solve the problem within the context you are given. Because we are trying to make it fit our equation, we assume that a nonlinear phenomenon is linear. But then we aren't solving the greater problem—the kind of problem that I like to give to my design students.

I can't tell analytical engineering students, "The problem is drunk driving."

What is the problem I'm giving them? There's the point of view of the Mothers Against Drunk Driving, there's the point of view of the cops, and there's that of the bartenders (not to mention that of the drivers). What the problem is depends on your point of view. If you say that the problem is drunk driving, engineers quickly get into fixing the car so that you can't start it if you're drunk. That's fine, as far as it goes, but, for the designer, the issues go further, and aren't so easy to formulate.

The designer can handle the messiness and ambiguity, and is willing to trust intuition. Basically, design has to do with intuition. Engineers are prone to assume that intuition does not exist—that you can't make any creative leaps without proving the solution through use of some equation.

BH: And yet good engineers seem to have a strong underlying engineering intuition about what kinds of solutions should be considered.

DK: That's experience. You feed your intuition by having plenty of experience. Good engineers are really designer–engineers—designers in every sense of the word. I'm just saying that the engineering profession has a self-image that it is based on mathematics and certainty. It justifies its status that way.

BH: You suggest that a designer's approach to a problem situation is dramatically different from that of an engineer. Is that a result of training, of temperament, or of experience after training?

DK: It's certainly a matter of training. I have had good luck taking artists and turning them into designers in the engineering–design sense. They learn the technical base, such as being able to calculate what kind of load a strut can take. Artists, because of their training, have learned to be more open ended, rather than problem oriented. Their training includes "Do something with this clean sheet of paper." They’ve learned to do something cool. Sculptors, in particular, seem to take well to becoming engineer–designers.

Everyone has the potential to be creative. The proof of that idea for me is the creativity of little kids. Little kids all draw. The refrigerator in a kid's house is full of drawings. Somewhere along the line, some teacher says, “Johnny, that’s not a very good picture of a horse. Billy’s picture is a good picture of a horse.” Then it’s all over—Johnny doesn’t draw anymore. Kids are inherently creative. Somehow, some designers have been allowed to have that training that allows creativity to show up, to get away with it, to know that their horses are different but are just as good, if not better.

Engineers have been trained to follow a methodology. Whatever comes up, they are supposed to apply the methodology—to analyze and synthesize. At Stanford, we have a program in product design that accepts engineers and tries to move them beyond this methodology. First, we loosen up the students—make them improvise, make them take risks, break cultural sets. People who are willing to loosen up can make great designers, but they have to be willing to take a chance.

When I was a student in the sixties, I took a class where they blindfolded us and had us walk around and feel leaves on trees. Trained engineers have trouble doing that kind of thing—they get upset. “What are we doing? Why are we wasting our time?” They question the value, instead of going with the experience. Attitude is a result of training; but you can cross the line afterward—you can recover.

BH: In developing courses on design, it has struck me at times that the most you can do is to create a situation that allows students to learn for themselves. You can set up semistructured experiences, you can mentor them, you can help them to become reflective about what they’re doing. But you cannot tell them what to do and have them know it—that’s just not the nature of design.

DK: For me, the important thing is to set up an environment that makes it okay to try things and explore. Every experience that students have in school pushes them in the opposite direction. We want to say that it is okay to fail, it is okay to try something that doesn’t work out. We're going to reward a spectacular failure in the same way we reward a success; that's not true in most situations.

Another aspect of training that comes up these days is the importance of hands-on experience. We have classes at Stanford now in the mechanical-engineering department on *mechanical dissection*. Twenty years ago, you didn't have classes in mechanical dissection. All the kids coming through the mechanical-engineering program had already taken apart cars and appliances, and had a toolbox full of tools in the car that they could use to dismantle things and fix things. Today, students in mechanical engineering know the subject from textbooks. Their hands have never been soiled, so to speak. So, we need to get them back in touch with the physical phenomena. They've built programs on the computer. That gives them a sense of satisfaction of creation, but it's different from mechanical fiddling. They don't get the feel of using a screwdriver to tighten a screw—of having it resist turning, or slip off. The physical experience gives a different feel for what happens with mechanical devices.

BH: What do students learn about design through these experiences?

DK: The term *design* can be misleading. When I say "I'm a designer," people sometimes ask me what I think of the color of their drapes. Their image of a designer is Calvin Klein. Even when you say "product design," the word *design* is a problem.

I like the fact that *design* has such a broad usage—that it is used in so many different contexts and is cross-disciplinary. A narrower definition might be more comfortable, but it would not be accurate. In my opinion, design in companies ought to be like marketing or manufacturing. We now have engineering or product-development departments with their own vice-presidents and official roles in the process, but we don't have design departments. I think we should (see Don Norman's account in Chapter 12). Design defines what *it*—whether *it* is drapes or software—ought to be. By contrast, engineering *does* it. Engineering is implementation.

Design has three activities: *understand, observe,* and *visualize.* Remember, design is messy; designers try to understand the mess. They observe to how their products will be used; design is about users and use They visualize, which is the act of deciding what *it* is.

BH: What you mean by "deciding what *it* is?"

DK: If I'm trying to design a tape recorder, I can create one that's yellow that you can dunk in water; I can create one that's highly precise; I can create one that has 27 heads or scads of features—that's what I mean by "deciding what *it* is."

When we think of software designers, we picture them in front of the screen. When we think of product designers, we think of them making prototypes. When we think of fashion designers, we picture them cutting out the fabric and making a dress. The key point of design takes place before any of those activities occur, and it requires making an uncomfortable leap—uncomfortable even for the most fluid and flexible people. This leap is the act that needs to come before implementation.

BH: Why is this leap so uncomfortable?

DK: A good problem in design is one for which you aren't sure that there *is* a right answer. In engineering, you might set out to make a tape recorder that meets certain performance or cost specifications. You can feel comfortable about making that tape recorder; you might be able to meet those specs, or you might not, but you're not taking a big risk. The risk is a technical risk, which is not the same as a design risk. When you're doing design, you're making a decision to create a thing, and you don't know who might say "That's the wrong thing." It could be a marketing person, it could be the user. You have to make the leap first, and you can't feel comfortable about the leap because it's too uncertain.

BH: What does it take to leap well?

DK: Who's good at leaping? People who have confidence. It's not that they are *comfortable* with it; they have just somehow been anointed with the ability to make this leap, and nobody is arguing. Look at a Steve Jobs; look at how he started Apple. Somehow, during his life, nobody ever successfully discouraged him from having grandiose ideas about what was possible. I've been with him many times when he made a huge list specifying what a product was going to be, and the list wasn't based on anything that was actually possible. He would say, "Oh, a disk drive, that'll cost about six bucks, so I'll put six down for that." Disk drives don't cost six bucks! But nobody ever discourages him from thinking he can do it.

BH: It's kind of a bravado that to another person could border on lunacy.

DK: Well, think about inventors. All my heroes are inventors, starting with da Vinci. If I had to pick one person who was an extreme, who could make the creative leap to what it ought to be, who could dream up things that didn't exist, he would be my choice. People who are good inventors—good designers—don't mind saying, "How about following this path, which doesn't yet exist?" Most people have trouble doing that; they preface everything they say in a brainstorm with "This idea may be stupid, but here it is." They are afraid that someone else will think that their leap is stupid.

Successful designers just send out their vision to the world; and then, when somebody else builds on it, that's okay. They're not protective of their ideas because they're so used to having ideas. A creative designer has an idea a minute. Publicizing an idea is a way to improve on the idea—someone else can build on it, expand it. If you're fluent with ideas, as most design people are, you don't have to be fearful. You don't protect your one good idea because your afraid you'll never have another good one.

The other point about making the leap is being able to keep seeing a problem from new perspectives. Anybody who has a strong filter on the world forces everything to fit that filter. In our product-design courses, we assign exercises where you take an object and look at it backward, or you draw it from an ant's point of view, or see it as material. For example, think about a computer monitor. It is just plastic with a piece of glass.Maybe I could do something completely different with it.

BH: So design requires inherent flexibility. We need to let objects and ideas assume new aspects depending on their context—we question an object's already-defined purpose or an idea's previously determined meaning.

DK: A central tenet of design is that you have to avoid stereotyping—the designer has to continue to say, "That's interesting! Look at that!" That skill is hard to teach. All you can do is to give people the experience, and to let them extrapolate from it. A good designer has always had substantial experience. If you ask designers "What's design?" they can't tell you. But a designer who has had experience knows what design is.

BH: How do you organize your company to foster creative design?

DK: Successful design is done by teams. Creative leaps might be taken by individuals, but design thrives on the different points of view found in teams. You want a multidisciplinary team, what we call *x-func* (cross-functional). You want different brains working on the problem. Otherwise, the person with the power, or the person who speaks the loudest, sets the direction for the whole design. You have to keep a broad perspective at the beginning; you have to be uncomfortable with the fact that you're not moving forward in a straight line toward the goal (and, project managers *do* want to move forward!); you have to feel comfortable with exploring.

It is important to recognize that there are a zillions of possible solutions, rather than just one solution. You look at the universe of possible solutions by making use of different brains, considering different functions, different users, and different manufacturers. After you look as broadly as you can, then you can feel more comfortable narrowing.

If you look broadly, your solutions will be completely different from one another. One of them may be based more on the social issues than on the performance and technology; one may be based on focusing on the user more than on the technology. To get all these points of view, you need a huge team with a broad base.

Later, you can use fewer people on the team, because you know where you are heading. As you narrow the focus, you know better what you need and what you don't. The appropriate people remain interested in the problem, and it continually narrows down as you move toward implementation. It's important to have fewer people as you make decisions. A bigger group can't get anything done; the real design still takes a creative leap. A group can't easily take the uncomfortable leap.

BH: How does the narrowing down process get started?

DK: That's a good question. There are two ways of moving a team forward—through a leader, or through a decision maker. In the first case, somewhere in the process, a leader naturally emerges out of the group—what I call the *strong, fair leader* with a strong personality who is willing to lead. This leader is not necessarily the person who set up the team in the beginning—the leader has to emerge naturally. That person will be focused on the solution for the right reasons, allowing the design to go wherever it goes, without injecting a personal bias. Such a leader doesn't have a stake in what specific direction the project goes, but just cares about facilitating its progress.

The alternative is what we call *user–chooser.* You build a customer council of users that is strong enough to make significant decisions about the product.

BH: Outside users?

DK: Yes, the potential users of the product. If I'm developing a toaster, I get 25 insightful homemakers, or a cross-section of people who make toast, and I present ideas to them. I give them several choices and ask, "What do you think?"

Using either approach, you can still make mistakes. The learning has to continue all the way to the end. The driving question is how you narrow down the alternatives—that's the uncomfortable part. Coming up with all the wild ideas and possibilities, that's not uncomfortable—that's fun. Asking the question, "How do you narrow?" is the same as asking, "How do you make the leap, given all the choices?"

BH: Again, it sounds like that is the kind of thinking that requires intuition, or experience.

DK: Yes, experience provides a basis that isn't the same as a set of methods or rules. When I get a concrete design question, I love to answer in the form, "It's been my experience that...," because no one can argue with that—it's just my experience. I can't offer clients that a comprehensive set of rules or a theory from which I can conclude, "Client: You should do this." But that makes them uncomfortable, and they say to me, "Oh, I thought you were good." Or "I thought this was your business." So, I say, "It has been my experience that this is what you should do."

BH: Clearly, experience is central, but can it backfire? Can you have too much experience, such that every situation seems to be a minor variant of a situation you have already seen and understood—so you don't respond creatively to what is, in fact, always a new situation?

DK: That's an interesting phenomenon. Perhaps it's a sort of a laziness in humans. Once you get to a certain point, you say, "I know that subject," so you stop collecting information about the subject. Then, there's another kind of person, the expert. The expert doesn't have that mentality at all. I don’t understand such experts, I just love that they exist.

For example, a concert pianist who plays a piece spectacularly well, after taking a year to learn it, still wants to improve the performance to that next level, not being satisfied with near perfection. A virtuoso isn't even satisfied with perfection. Experts don’t fall into thinking they know it. Even though someone may be the top expert in optics, or the most accomplished pianist, she hasn’t fallen into that trap of thinking that she knows it.Experts are really narrow and really deep. That’s different from a designer's mentality.

BH: In some environments, there can be a tension between *my* creativity and *yours*. In other words, there may be a certain group of people who are expected to be creative, effectively precluding other team members from full participation. How do you avoid this situation?

DK: In a company such as IDEO, where you have many creative people, when someone has a great idea, nobody ends up knowing whose idea it was. It’s not branded as *George’s idea*. Either nobody knows whose idea it was, because of all the discussion that took place, or everyone thinks it was his or her own idea. Either perception is equally good, and both happen frequently.

In our company, the person who owns the solution is the person who posed the question. If I’m the one who says, “Let’s create a new toaster,” I own everything that happens, because I initiated the discussion. In most cases, the clients are the owners, because they pose the question. You can hold a brainstorm on any subject you want, any time. Just send out a message that says, “I’m holding a brainstorm at 2:00. Will you come?” And everybody comes. You’d think everybody would say, “Oh, I’m too busy.” Not true: People know that they are going to need you to come to their brainstorm tomorrow. So the culture requires you to participate in other people’s problems.

You eventually get to the point where you look forward to the brainstorm for two reasons. One is that sessions are genuinely interesting. It’s like reading a book or reading the encyclopedia—you learn about a new subject. The other reason is that it’s not your problem. The guy who’s got the toaster problem has the client; he has to get the design done on time; he has to make that leap that makes us so uncomfortable. The other designers come in totally relaxed, totally free to generate ideas in the toaster area, because it’s just fun. They’re helping, and the only pressure on them is that they want to help their peer.

BH: You say that clients are often the owners of the problems. How much initiative do you take in helping them to define the problem?

DK: Say that a company comes in and says “We want you to design a new toaster.” And I say “We’re going to study bread crisping in its generic sense.” They say, “Look, I told you just to design a toaster. Get going.” They think that the world of what a toaster could be is small. But we might say, “Look, our job is to begin by looking at the history of bread." Maybe we want to see whether they had a better way to make toast in Renaissance France or on the Space Shuttle. We don’t want just to design the curlicues on the side of the client's next toaster. Maybe what we’re going to find from looking at this history is that the best solution is to put more curlicues on the side, but I always say: “Think about how much more comfortable you’re going to feel that we are doing the right thing if we do a broad search and then narrow down.”

Breadth takes extra effort at the beginning, but that's the best time—right at the outset of the product development process—to take a step back to see the situation. The design part of the product process takes a small portion of time. The next steps after design have 10 times the cost, time, effort, and emotional stress. Then, manufacturing incurs 10 times that. So time invested at the beginning isn’t wasted; we make up the cost in later stages easily, by doing the design right.

BH: It seems that there is an implicit tension when you do design in the real world. On the one hand, you’ve been saying that the design process is inherently unpredictable and you have to be willing to stand naked. You don’t know where an idea will go, and you don’t know how long it will take to get there. On the other hand, organizations, by their nature, once they reach a certain size and complexity, have a strong need to know—or at least they think they need to know—what the future will be like. How do you balance the uncertainty of design against this need for certainty?

DK: You can't put design in a structure. You can't give a company a methodology manual. If they are unable to accept uncertainty in answers to “When is it going to be finished?" and "What is it going to cost?, I always say to them, “How long does it take you to come up with a good idea?” Of course, that doesn't mean that we should stop trying to be mindful of our process when doing design. Every company should try to extrapolate from what has made a certain design successful—to operate in learning mode.

BH: Can a company learn how to make design projects work?

DK: Let me give you an analogy. Jim Adams, who’s a professor in our group at Stanford, writes books on creativity and learning (Adams, 1974; 1986). He uses the example of learning to tie your shoes. When you’re a kid and you don’t know how to tie your shoes, there’s no tension, because you aren’t aware that other kids do know. Then, you get to the stage where you know that you don’t know how to tie your shoes, so you want to know. The thought of learning how to tie shoes moves into conscious awareness. The prior stage is unconscious.

You’ve got to be conscious. Companies can’t stay unconscious, and say, “We don’t know how it happened, so we’ll just hire some good people.” Once you know that you don’t know, that is the opening for all the learning to occur. That’s the stage where you’re trying to figure out how to tie your shoes, and you’re alert when you see other people tie their shoes. You watch.

Then, you get into the next stage, where you have learned recently how to tie your shoes, but you’re still mindful when you do it. That’s also a conscious state, a good one. Then, you get to the stage where it’s automatic: You don’t think about it, you just do it. I probably couldn’t teach somebody else how to tie shoes, because I don’t think about it—I don’t remember how I tie my shoes.

You can’t quit trying to learn about the process and stop being mindful of it. Companies need to keep striving to understand and improve their design process if they expect better results. But that is not the same as expecting that study will lead to standardizing the process. You think that you’ve understood a problem, but then it turns out that you cannot solve the next problem in the same way. The typical design situation requires doing something that you don’t yet know how to do.

BH: That last phrase echoes what you said at the beginning: that the designer has to dream beyond what exists, as opposed to fixing what exists.

DK: Yes, that is the essence of design. What drives my life as a designer is ultimately the joy of creation, of doing something that works, of working with other people. It's the thrill of hitting that inner need to create, along with that outer need to satisfy other people and to make things go just a bit more smoothly and a bit more elegantly in the world.

Suggested Readings

James Adams. *Conceptual Blockbusting: A Guide to Better Ideas* (3rd edition). Reading MA: Addison-Wesley, 1986. (First edition, 1974.)

James Adams. *The Care and Feeding Of Ideas*. Reading MA: Addison-Wesley, 1986.

Don Koberg. *The Universal Traveler: A Soft-Systems Guide*. Los Altos, CA: Kaufmann, 1974.

Don Koberg and Jim Bagnall. *The All New Universal Traveler*. Los Altos, CA: Kaufmann, 1981.

About the Authors

David Kelley is a Professor of Mechanical Engineering in the Design Division of the Mechanical Engineering Department at Stanford University. He is also CEO and founder of IDEO Product Development, America's largest engineering and design consultancy.

Bradley Hartfield is a software-design consultant and has been a lecturer at Stanford University, where he has played a major role in developing Stanford's program in Human–Computer Interaction Design.

Taken from: https://hci.stanford.edu/publications/bds/8-kelley.html