Title

Design for Use

Journal

Research-Technology Management, 59(1)

ISSN

0034-5334

Authors

Norman, Don

Euchner, Jim

Publication Date

2016-01-02

DOI

10.1080/08956308.2016.1117315

Peer reviewed

Research-Technology Management

ISSN: 0895-6308 (Print) 1930-0166 (Online) Journal homepage: http://www.tandfonline.com/loi/urtm20

Design for Use

Don Norman & Jim Euchner

To cite this article: Don Norman & Jim Euchner (2016) Design for Use, Research-Technology Management, 59:1, 15-20

To link to this article: http://dx.doi.org/10.1080/08956308.2016.1117315

Design for Use

An Interview with Don Norman

Don Norman talks with Jim Euchner about the design of useful things, from everyday objects to autonomous vehicles.

Don Norman and Jim Euchner

Don Norman has studied everything from refrigerator thermostats to autonomous vehicles. In the process, he has derived a set of principles that govern what makes designed objects usable. In this interview, he discusses some of those principles, how design can be effectively integrated into technical organizations, and how designers can work as part of product development teams.

JIM EUCHNER [JE]: You have studied how people use designed things and how they get confused about their use. What principles underlie the design of useful objects?

DON NORMAN [DN]: Designing appropriate technologies is far more complex than most people realize. There is a tendency to focus on the technology itself rather than on the context of use.

There's also a tendency to think that we understand the way the technology is going to be used. But neither of these assumptions is true. The real issue is not technology but the benefits that a design provides to users. And, of course, we are designing for other people, not for ourselves.

We need to understand that we don't inherently understand other people. Even if they are like us in every single obvious way, they are unlike us in one very special way. Once we start designing a product, we understand it in ways users never will. We've thought about the object, we understand its issues, we understand the compromises we're making. What we cannot seem to understand is that the people we're designing for not only don't know about all of these nuances; they don't care. They don't care how much effort we put into it, or the kind of choices we made, or the wonderful technology behind it all. They simply care that it makes their lives better. This requires us to do a design quite differently than has traditionally been the case.

JE: What are some of the principles that can help?

DN: First, make sure you're designing the right product; make sure that you're solving the correct problem. Someone said, "Doing the wrong thing correctly simply makes you wronger." So the first question to ask is, "What are we trying to do?" Quite often, we have an existing product, and we want to make it better; at other times, we're trying to keep up with the competition, which might have three more features than we do. We assume that we had better add those same three, plus two more, so that we're ahead.

That makes the product worse for everybody. What a designer needs to do is to create a cohesive understanding of what the product is about. The designer needs to reject anything that is not part of the essence of the product, to make sure that it stays focused. What you ought to emphasize is your strengths; if you focus too much on keeping up with the competition, you lose your distinct advantage—your product looks just like their product. Don't let this happen. You want to be distinctive, you want to focus on the things that you excel at and make the things that are core to your concept even better. When your competition introduces features that are not part of the essence, who cares?

JE: Who does this well in the physical space?

DN: It's hard to come up with examples. One of my favorite companies is Crown. They make lifters and forklifts—machines you see running around in warehouses and factories. But they make them really nicely. They are designed so that they are easy to use; they're also very attractive, so they're kind of neat to be driving.

Let me talk about how you create such products. You really need to start by understanding the people who are going to use the product. Don't send out questionnaires, please. Don't even interview the customers. Send out

Don Norman is director of the Design Lab at the University of California, San Diego, cofounder of the Nielsen Norman Group, a member of the National Academy of Engineering, an IDEO fellow, and a former vice president of Apple. He helps companies make products more enjoyable, understandable, and profitable. His books include *Emotional Design*, *Living with Complexity*, and an expanded, revised edition of *Design of Everyday Things*. He lives at www.jnd.org.

Jim Euchner is editor-in-chief of *Research-Technology Management* and vice president of global innovation at Goodyear. He previously held senior management positions in the leadership of innovation at Pitney Bowes and Bell Atlantic. He holds BS and MS degrees in mechanical and aerospace engineering from Cornell and Princeton Universities, respectively, and an MBA from Southern Methodist University. euchner@iriweb.org

DOI: 10.1080/08956308.2016.1117315

Don Norman helps technology companies integrate design thinking into their innovations.

people to *watch* the way customers do their work. They may have problems that you will observe, problems that they aren't even well aware of, problems that the customer has always felt are just the way it has to be. You get your competitive advantage by eliminating these problems.

Look for instances where people are using a product in ways that you never dreamed of. Look for subtle signs of a problem with the use of the product, like taping over a button or putting signs on top of a device to remind people how to use it. Try to notice where users have taken two or three other devices and tried to use them at the same time as they are using your product, just to make it possible to do their work. Or where they're using your product for a task you never envisioned it being used for. All of those examples are wonderful because they give you opportunities to bring out something new that really enhances the work done with your product.

JE: In your writing, you talk about basic design principles, things like visibility and feedback and mapping, and engagement or joy. Can you talk about the principles that people should think about as they're doing the design, once they've completed some of the customer insight work?

DN: There are fundamental principles of design that I have covered in many of my books: make use of the product clear so that people can discover what they can do with it, which is called *discoverability*. If a user does something inappropriate, help them to recover—that's *recoverability*. Help users understand what's going on when they're using a product through *feedback*, so they know when something has happened.

Try to create a good *conceptual model* for the user, one that helps them understand the device. People don't need those conceptual models when everything works, but when something goes wrong, when there's an unexpected reaction or when people have accidentally done something that gives an unexpected result and they have to recover, that's when it's essential that people have a good mental model of how the thing works.

JE: Is there something that you would advocate beyond what might be called traditional design methods?

DN: First, when I lead teams, we go and observe people doing their work. Afterwards we talk with them. Sometimes, after we make our observations, we'll show them what we observed, and ask what they were doing, what they were thinking, what their intentions were. But during the actual activity, we mainly observe.

Next, we then take those observations—we usually do this as a group—and try to understand how they can be categorized and how they might be combined. We ask what opportunities might be present in that world.

Then, we decide which opportunities to focus on. We might do a very quick test on the people we observed—whip out a prototype, usually in hours, one that just illustrates the concept. We go back to our sample people and ask them to make believe that they're using our prototype. That gives us rapid feedback about whether we have the right idea in the first place.

People don't know what they want. That's why it's important that we do this early work effectively, that we build a prototype, test it, and discover what the true requirements are. And then we do it again, and again, and again, each time, learning more and more what the concept needs to be.

JE: That's a useful design process. What is your experience with this as you move more towards more purely digital products? Is it harder to design a good digital product?

DN: Digital technology increases the power of objects, but it can increase the difficulty of design, as well. The power of digital devices and displays is in their increased flexibility, but if you overuse that flexibility—when you give people too many choices to choose from—they often just give up.

None of us can manage a device if there are too many choices to be made; we just want to get our jobs done. Moreover, if we give everybody the ability to tailor a product for their own special needs, most people won't bother; it's too complex. And if they do, it makes it really hard on service people. It's easy to be tempted, but it's not always productive to use the flexibility digital devices provide.

Now, most jobs are complex, so the device has to be complex. People claim they want simplicity, but if you make things simple, then they complain that it doesn't do all the things they need done. The complexity of the device has to match the complexity of the world. But let's distinguish complex from confusing: it is confusion that people dislike, not complexity.

Fortunately, if you create a good conceptual model, people won't think it's complex; they will think it's just fine.

The flexibility given to us by digital technology can be useful during the design process because it allows us to do quick tests. We can mock something up really quickly and make the device look like it's actually working. This lets people experiment with it, push the buttons, and see

the screens change. It looks like it's working (even though it isn't), so designers get feedback quickly. But don't let the flexibility of the prototype bleed into the product unless it's really needed.

JE: You're arguing that designers would be better off sacrificing a little fidelity for speed at this stage.

DN: Absolutely. The quality of insight from these prototypes is really valuable. I have heard people argue that this information is just anecdotal: that it's based on just 30 people (we actually recommend using just 5 people—faster that way). Marketing department surveys have thousands or tens of thousands of respondents. I'd argue that the surveys take too long to do and that I don't trust their results. Come on folks, do you trust the answers that you put down for a survey you've been asked to fill out? No; so why trust a survey when you are designing a product?

Don't try to cover the entire population with your research; try to get to the crucial people—the people who are the outliers. Don't test the average person; test the super-great person and the super-bad person, the hypercritical person and the sloppy person. Test around the edges; if you can satisfy those users, you can satisfy a lot of people in between.

JE: I read something that you wrote about procrastinated binding in the design of digital products. My understanding is that procrastinated binding allows for the user to adapt the product for use after it has been manufactured. How does that fit with a human-centered design approach? How does a designer keep the user in mind instead of focusing on the platform strategy?

DN: That's a very complex issue. It's very easy to be flexible in the early days of a novel product—we can keep our minds open and try to design the product for a range of alternative uses. But this gets more difficult over time, as you get further along. Eventually at some point, you actually have to determine the architecture. The architecture and underlying infrastructure can take a long time to build. And once you commit to it, you have a platform that's very difficult to shift.

Finding the proper architecture is probably the most important thing in product design. It's interesting that many companies don't focus on this; they focus on the first product or two and then try to figure out what the underlying architecture should be. Then it's too late; the pressure will be on to do a third product, so the architecture is never built properly. Do it correctly at the very beginning because there will never be another chance. But if you don't focus on the correct architecture from the beginning, the result will not sustain future growth.

JE: So getting the architecture right is critical; it's also very difficult because it's fairly abstract, right?

DN: That's why it's really important that the design community participate from the start. The tech community might say, "This is all technology. Why do we need you?" But one of the worst things that can happen is that, several years into the product cycle, the customers are asking for some new kind of capability that the architecture can't deliver. Designers can help prevent this.

JE: Let's talk about another area that you've worked in and that is rapidly emerging—autonomous products. You've done work with autopilots on planes and with autonomous automobiles. It's paradoxical, but usability is critical in this space as well. What are the special challenges as equipment becomes autonomous?

DN: I think that autonomous vehicles would be great; I like the things that they could deliver. But when something is only 99 percent autonomous, that's dangerous. There are two different issues here: One is the danger of partial autonomy. The other is the philosophy for designing autonomous equipment and automation.

Let me talk about the philosophy first. There's a tendency to automate anything that can be automated. Automate whatever can be automated and leave the rest for people. Other than replacing people's jobs—which is a major problem for the nation—it doesn't always work well. This partial automation means that people are picking up what machines can't do. People are asked to behave on the machine's terms, not the other way around, and there can be counterproductive results. People are not machines; we are not good at high precision or high accuracy; we are not good at boring repetitive tasks. And when we are asked to do things we are bad at and make errors, we are blamed; human error is the culprit. No, this is design error: asking people to do things they are bad at.

If things are designed with the human in mind, the combination can work well. People are not good at precision, we're not good at repetitive operations, and we're not good at monitoring something when nothing is happening—we're not good at being asked to intervene unexpectedly. We're very good at creative things, though. We're good at formulating problems. We should strive to automate what people are not good at, and try to use technology to make the things that people do even better.

Want a good example? Consider the calculator. Calculators don't work the same ways that we do when we do arithmetic, algebra, or even calculus. It's a good thing they don't because we make errors and the calculator doesn't.But the calculator can't formulate the problem in the first place. Once we have formulated the problem, the

Let's distinguish complex from confusing; it is confusion that people dislike, not complexity.

calculator can solve it accurately and efficiently. The calculator helps us formulate the problem because it gives us quick answers that indicate whether we are on the right track.

The calculator is a machine that adds to our intelligence. It takes away what we're *not* good at, and it allows us to fully focus on something we are good at. In many ways the word processor does the same thing; we can get our ideas down quickly and worry about the formatting and the other stuff later. But whenever we use technology blindly, we have people doing what they're bad at; then, when people do what they're bad at, and they do it badly, we blame the people—it's "human error." All we have really done, though, is to learn that people are people.

JE: I've noticed, as well, that this tendency to automate whatever you can automate badly fragments work. None of the people doing the remaining work has the context to do much more than a task. But many problem-solving situations require a much wider context than a task if they are to be resolved well.

DN: Absolutely. A particular instance of a job that might not be done well in a partially automated world is safety. Take automating the piloting of planes or the driving of automobiles. There are a lot of advantages that would result from complete automation of the automobile. Currently, human-operated vehicles injure a million people a year, and kill more than 30,000 people in the United States. People are not really good at driving. We forget how difficult it was to learn because once we become skilled, it can be done automatically, without much conscious awareness.

What do people do when they get complacent about driving? They try reading a map or an app while driving. They drop stuff on the floor and bend over to pick it up. They answer the phone or change the radio settings. At 60 miles an hour, we go 90 feet in one second. Yet the distractions of dialing the phone, selecting the radio station, or even looking at a passenger can easily take 10 or 20 or 30 seconds. And then, if some unexpected situation arises, it arises quickly, requiring an immediate response. We're not good at that.

I would love to see *fully* autonomous driving. The problem is partial autonomy. Planes can pretty much fly themselves most of the time. And when a plane gets into trouble, we can rely on highly trained pilots, who have been trained in extremely sophisticated simulators, to take over. Even so, when the problems do occur, it can take a minute or two, or more, for the pilots to figure out what's going on.

They usually have the time. They're way up there, at 40,000 feet or so, and they have several minutes to think things through; even so, when something suddenly happens, they're often completely unprepared for it, no matter how well trained they are. Most people can't sit in an airplane for hours and hours at a time, with everything working perfectly, and then suddenly respond when an unexpected event happens.

For an automobile, by contrast, we have untrained drivers who might have only a second to respond. If the autonomy cuts out for some reason, and the driver has become distracted, there is no time to react.

JE: Have people found ways of keeping drivers engaged, even on a secondary task, so that they are more alert when they need to be?

DN: It's a solution people have considered, but the problem in practice is that, unless the secondary task that you're doing is meaningful, people just tend not to engage with it.

A possibility that I like is to engage people with high-level tasks. They are not driving tasks—trying to keep the car centered in the lane, deciding on the next turn—but higher-level tasks like directing the trip: "I want to go to the bank" or "I'd like a more scenic route." That creates engagement to some degree in what's happening but at a level that seems comfortable; it doesn't feel like make-work, and it may keep the driver involved so that when an issue arises, he or she can take over.

The second thing that can be helpful is automated advanced warning. Quite often, when issues occur, the automatic systems of the vehicle have been aware that something is reaching its limits well before the problem is made apparent. A system is trying to compensate for an unbalanced airplane, for example. Why don't the planes let the pilots know? This is a real story—I've seen this happen: the plane has automatic equipment that completely compensates for a problem and takes over until it reaches its limits; then it collapses. Goodbye. Instead of just compensating, why can't the automation notify the pilot of how much it *is* compensating and how close it is to its limits. That would be very helpful.

There are a fair number of situations like this. I've heard similar stories with automobiles, where a brake fails and the other three wheels compensate, using modern control systems. The driver never realizes that conditions are now a little more dangerous. When the second brake fails, if the system has not alerted the driver, it might not be noticed. But by then it is certainly time to tell the driver, "This is serious. You should drive really slowly or get off the road *now*."

JE: How fast do you think we'll get to autonomous cars? Do you agree with the people who think it's going to be 2020 or 2022, or are they dreaming?

We should strive to automate what people are not good at and try to use technology to make the things that people do even better.

Every engineer doesn't have to become a designer, but they have to respect designers.

DN: I flip-flop over this. I think it's going to be a long time before the technology is really fully ready. At the same time, I'm more and more concerned about the increasing level of distraction in today's automobiles, which makes autonomy more important. The vehicle as an environment for a new kind of entertainment and information system is innovative, and a lot of automobile manufacturers are implementing such systems, so driving in the future will involve more and more distracted driving.

I conclude that we've got to implement automation faster rather than slower, even before we have perfected it. Imperfect automation, in many instances, is going to be better than imperfect humans. We have reason to hope that, with time, automation will get better, and we know that, with time, people are getting worse.

JE: I'd like to turn to how we teach engineers about designing useful and usable objects and how we inform management about design so that they can manage it well in their companies.

DN: How we instruct designers, engineers, and manufacturing is different for each type of person. What we try to do is to teach in the language that they understand and talk in terms of the concerns that they have in the company. It's very akin to design itself.

For engineers, I focus on the challenge of user-centered design: it's harder to design something so that people can really use and understand it, so they often complain and say that what is being asked is impossible. I prod them: "You mean you can't do it? You mean you're not good enough to do it?"

Every engineer doesn't have to become a designer, but they have to respect designers. They need to see the designers as experts—not experts at programming or a technical discipline but expert members of the team: "They work together with us; they're equals with us. And they have skills I don't have, just like I have skills that they don't have." That's what we try to make happen in an engineering team.

JE: How do you help people on the team to communicate well about design? I've had teams that included designers and engineers and software designers; somebody or other's point view is often devalued, and it's often the designer, for various reasons.

DN: Sometimes it's the designer's fault. In fact, often it's the designer's fault because they don't seek to understand the problem of the programmers. I think designers should learn some programming to understand how hard it is, and programmers should take at least a short course in design to understand its powers and difficulties. But you're right, any time you mix cultures, you get difficulties.

JE: Have you found anything to be particularly useful in overcoming those difficulties?

DN: The project directors are the key people. They're the ones who see the whole picture, and they're the ones who can help to inspire people. You only need a couple of senior people—senior designers, senior programmers or engineers—to embrace this, because they will help the other people. It's not easy to manage this, because it can come down to a personnel issue; sometimes there are people who are just bad at this team approach, and they shouldn't be part of it.

When we are trying to convince management about the importance of design, we designers have to talk in managerial terms—in terms of costs and increased sales. Most designers, when I tell them this, say, "Well, how can we do that? We don't know what the sales will be." I tell them that this is what marketing people do all the time. Get them on your side. Talk to management by convincing them that design and usable systems are beneficial to the company. Instead, most designers focus on how important usability is, or how important it is that users not be confused. Everyone agrees with that, but that's not the kind of talk that helps senior management. Senior management is rewarded by increased sales, margins, and profits, so we must explain how design contributes to those outcomes.

I don't have any special methods for making this collaboration work. I will say that some companies, for example those started by engineers, or who are engineer-driven, probably never will understand design. To name one, look at Hewlett Packard. The curse of HP, as well as its past success, is that it was an engineer-driven company. Even though lots of people have tried for many years to change the culture—they even had a Chief Design Officer for a while—it never stuck.

Culture is the most important component in making this work, and culture is most difficult to change.

We are now in the midst of an exciting technological revolution, one that is not so much about the technology as it is about the way that people and society interact with the technology—or better, how people interact with one another, mediated by technology. This means that people's needs, behavior, and capabilities are the driving forces. Technology is an enabler but not the driver. And so to enable the wonderful opportunities for social engagement, for a more enjoyable productive work environment for more effective learning, and for exciting entertainment, we need to understand people. We need to design for people and with people. Technology and people need to interact seamlessly, joyfully. Designers, engineers, and business people all need to collaborate as equals.

The Industrial Research Institute and the Kellogg School of Management present

SHAPING INNOVATION LEADERS

THE PREMIER EXECUTIVE MANAGEMENT COURSE DESIGNED EXCLUSIVELY FOR TECHNOLOGY LEADERS

June 4-10, 2016

Northwestern University's Kellogg School of Management

Register by January 22 for substantial savings
Discounts available for multiple registrants
Visit www.iriweb.org for more details

NORTHWESTERN UNIVERSITY
Kellogg
School of Management

IRI Industrial™ Research Institute
Creating Innovation Leadership Solutions