ERIC REISS

Usable Usability

Simple Steps for Making Stuff Better

Contents

Part One
Ease of Use

These first five chapters are about physical parameters, which basically ensure that something does what you want it to do. Buttons, controls, and other response mechanisms are there to help you accomplish your task, and they might include functions and features that may even anticipate your needs and habits. In short, these things make stuff easy to use.

You might think that this idea is something of a no-brainer, but it isn't. Despite all the lip service to "user-friendliness," a depressing number of programs and products are still pretty UN-friendly. Throughout the next five chapters, I'm going to show you how well-meaning design doesn't always lead to well-functioning stuff.

What's in this part?

This part covers the following aspects of "ease of use":

- Functional (it actually works)
- Responsive (I know it's working; it knows where it's working)
- Ergonomic (I can easily see, click, poke, twist, and turn stuff)
- Convenient (everything is right where I need it)
- Foolproof (the designer helps me to not make mistakes or break stuff)

I have this goofy hope that when you see this list, you will say to yourself, "Yeah. That makes sense. What's the big deal?" But to illustrate my point, please take a moment to go to your favorite website. Click around for a couple of minutes while thinking about these issues. Can you see something that could be improved based on anything on this list? I bet you can! Welcome to the world of usability.

Chapter One
Functional

Flick a light switch and you expect the lights to come on. Turn the key in the ignition and you expect your car to start. You expect your refrigerator to be cold, your oven to be hot. These are all functional interactions. If things don't work at this very basic level, then it really doesn't matter much how beautiful a design may be. So what better place to start a discussion of usability than with functionality?

Please keep in mind that there will be some overlap between the discussion of functionality and other issues in the "ease-of-use" part of this book. For now, I'm concentrating on the "works/doesn't work" aspects of usability and design, although I revisit some of the things discussed here throughout the book.

Delicious Spanish dessert at a trendy Madrid restaurant. But how did they expect me to use the very round spoon to get in the corners of the very square bowl? I used my fingers to circumvent this amusing functional failure.

In one of the most technologically sophisticated environments—the modern airport—a chock of wood with a rope attached is still the preferred method for keeping planes parked correctly. An elegantly simple and highly functional solution.

The three keys to functionality

Just for a moment, consider the faucet on your sink. When you turn the taps, you expect water to come out. You want to be able to easily adjust the temperature. And if you want very hot or very
cold water, you don't want to have to run the water for a long time.

In more general terms, these same three functions also sum up the basic needs on a website:

- The buttons and links must work when you click them.
- The navigation must be responsive.
- The processing speed must be acceptable.

A frightening number of websites, apps, services, and so on fail for these very three reasons.
So do some faucets . . . the same generic issues guide many products in the physical world.

This frying pan is so poorly balanced that it can’t fry anything—unless you hold up the handle (or fry really heavy eggs). Functional deficiencies can appear in very strange ways; the average person looking at this utensil in a store won’t think to check the balance, so the designers should do it for them.

From click to conversion: making sure the buttons work

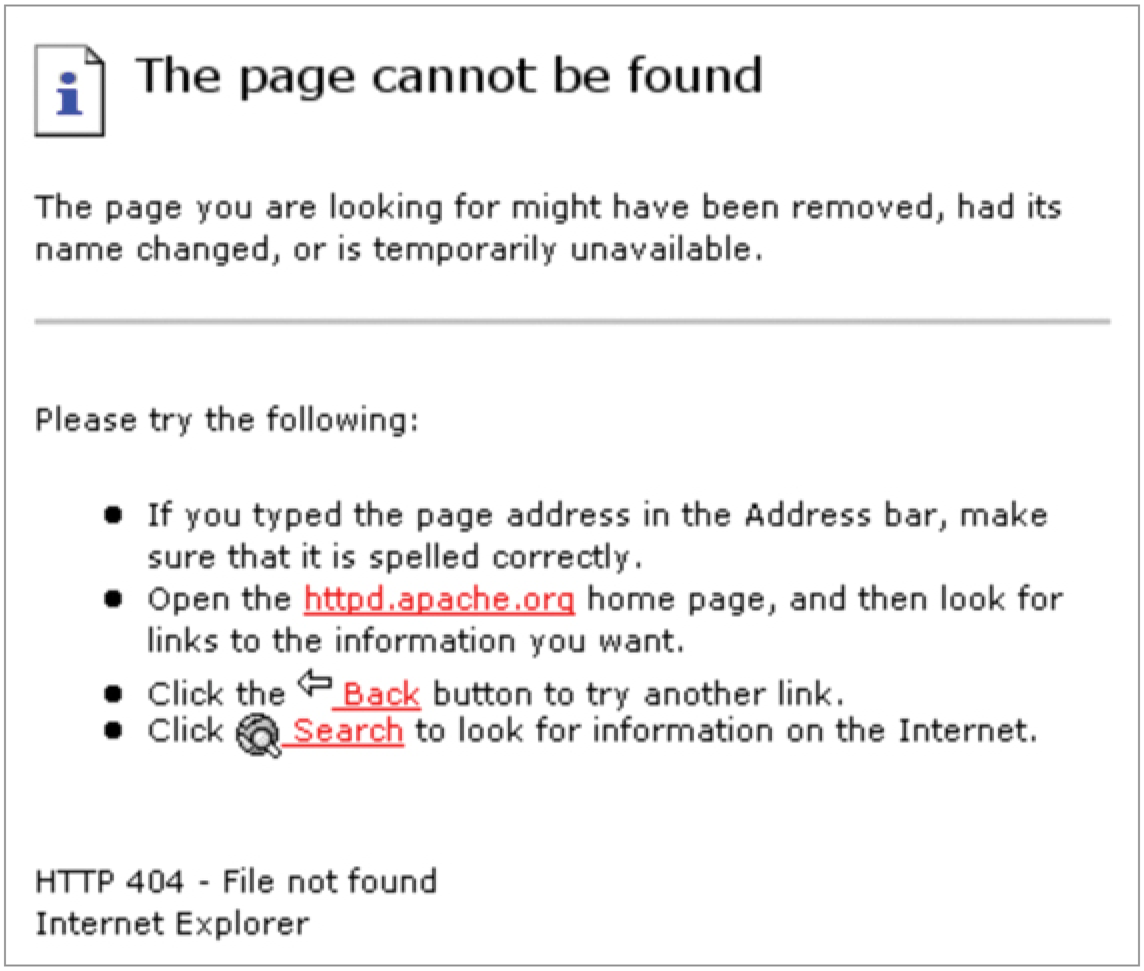
You’re probably thinking, “Hey, broken buttons should be fixed. This is a no-brainer.” And you’re right. Amazingly, broken buttons are a bigger problem than you may think.

And I don’t mean only links that may have broken, but also basic mechanisms that simply don’t work. Let me tell you a story.

My daughter-in-law wanted some earrings that were available on a jewelry store’s website. I found the earrings and clicked “Put in basket.” But when I went to check out, my basket was empty. Assuming I had made a mistake of some kind (because this problem was simply too absurd to show up on a professional e-commerce site), I repeated the procedure—and got the same useless result. Out of curiosity, I tried to put other products in my basket. Nothing made it to the check-out. Something was clearly broken.

When I called the store to order the earrings, I was told that practically all of their business is through their offline outlets. “Our sales via our website are pretty much nonexistent.” Duh. Of course sales are nonexistent if it is physically impossible to buy something, which forces people to call or visit the store in person.

Dead link? Server down? Something else? If your web analytics program is telling you that your 404 - Page Not Found error is getting a lot of page views, you need to investigate immediately.



The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Please try the following:

- If you typed the page address in the Address bar, make sure that it is spelled correctly.
- Open the httpd.apache.org home page, and then look for links to the information you want.
- Click the Back button to try another link.
- Click Search to look for information on the Internet.

HTTP 404 - File not found
Internet Explorer

As we spoke, I realized that the company had no clue as to *why* online sales were nil. Because the company didn't really take e-commerce seriously, no one at the company was actually looking very closely at the functionality of the website.

What *does* it cost a company when people can't deal with a company online? If no other channels exist (i.e. a business is only available online) then the cost will be significant, but then again, online-only companies probably look carefully at how the website is performing and catch problems very quickly. It's when alternative channels are available (a physical store, for example) that businesses tend to neglect the online presence, as the jewelry store did.

If the attitude of the site owner is "Well, we have a site because everyone else has one. . ." then you are bound to run into problems like this. Of course, these kinds of things are usually easy to fix, but you do have to find them first.

Browser wars, hardware headaches

Obviously, to check the functionality of any screen-based interactive product, the first thing to do is to click through it. For websites, various tools such as Google Analytics also help you spot dead links and such. But what you really want to look for are navigational elements that are programmed wrong so that they lead you to the wrong page, or even the same page (yes, this happens a lot).

You should also download a couple of different browsers to see that everything works equally well across a variety of tools. At a minimum, check your site in the following:

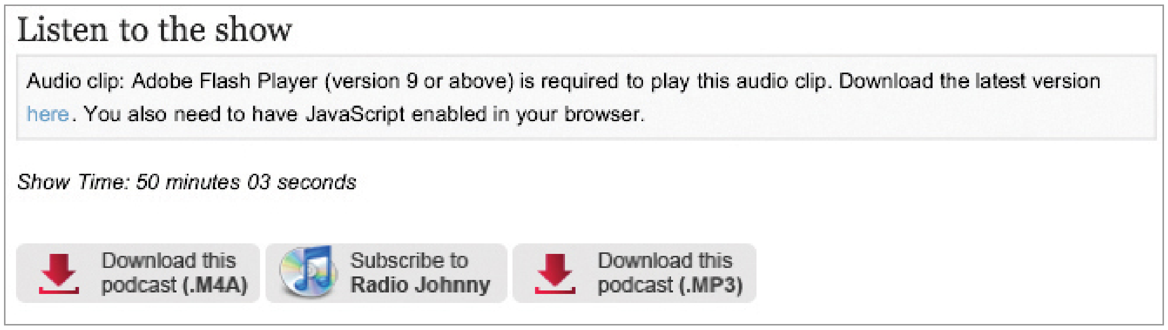
- Internet Explorer
- Safari
- Firefox
- Opera

You may also find that various small interactive elements, such as in-screen audio or video controls and animations, will simply not work across all platforms. For example, widgets programmed in Flash (an Adobe animation tool) will not display on some Apple products (notoriously the iPad[1]). If interactive elements are essential to your site, check their performance on popular operating systems for:

- Smartphones
- Tablets
- Laptops
- Smart TVs

[1] In the biography by Walter Isaacson (Simon & Schuster, 2011), Steve Jobs states that Flash technology used up too much of the iPad's precious battery power in relation to other programming options. Hmm. Legitimate technical consideration or business vendetta? The jury is still out.

On most devices, a small Flash graphic enables people to control an audio podcast (top). But an iPad does not display Flash, so these controls disappear, rendering the website unusable. Curiously, my iPad was nice enough to tell me where I could download the software it refuses to acknowledge (bottom).

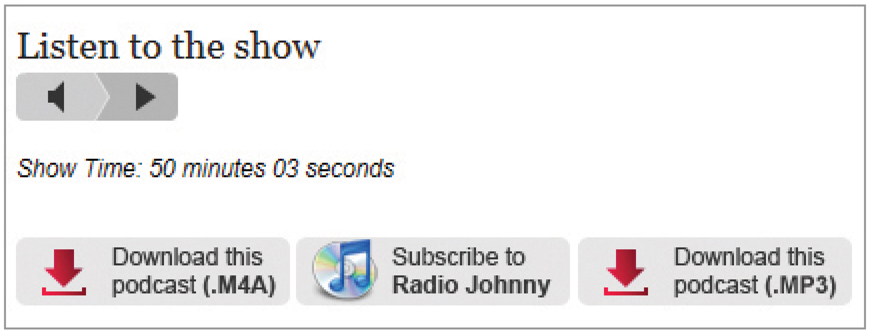


Listen to the show

Audio clip: Adobe Flash Player (version 9 or above) is required to play this audio clip. Download the latest version here. You also need to have JavaScript enabled in your browser.

Show Time: 50 minutes 03 seconds

Download this podcast (.M4A)
Subscribe to Radio Johnny
Download this podcast (.MP3)



Listen to the show

Show Time: 50 minutes 03 seconds

Download this podcast (.M4A)
Subscribe to Radio Johnny
Download this podcast (.MP3)

Don't sweat the home page. Fine-tune your forms.

I spend a depressing amount of time in design meetings listening to people fret about the home page of a website. Yet the home page is arguably the *least* important page. Sure, the home page gives you an opportunity to spell out the big picture—what the site is about—and display a range of informational/functional options available to the visitor.

But in truth, the better you design this online welcome mat, the less time visitors will spend on it. That’s because they’ll quickly spot the link that gets them where they want to go. Moreover, some people access your site from a search engine that leads them to a page far deeper into your site. Chances are that many folks don’t even see your home page.

From a business perspective, your home page probably isn’t where you create online conversions—which is almost always a top priority—getting people to order your product, sign up for your newsletter, download a document, submit a blog comment, or even just send you an e-mail. Conversions don’t always involve money (although many do). That said, most conversions require visitors to fill in an online form of some kind. Therefore, if you are going to fine-tune any pages on your site, you should concentrate on your forms.

Form problems are related to those of broken buttons because something on the site is preventing your visitors from interacting with you in the intended manner. However, as opposed to broken buttons that get in the way of *all* visitors, most form-design problems are much more difficult to spot because the form undoubtedly works for at least one group of users—the one on which the original design team focused.

Four keys to creating functional forms

Although I get into other aspects of form design later on, in terms of functionality, there are four things you need to keep in mind:

- People have to be able to provide the information you are demanding.

- Inflexible input formats greatly increase the chances of form failure.
- The need for interdependent forms and logins also increase the chances of failure.
- Instructions that are misleading are a great way to frustrate your users.

Obviously, there are other issues, such as the security of a password, how the on-screen messages are phrased, whether the layout is easy to understand, and so on. But one thing at a time. . . .

Required fields

A **field** is a section of a form—one of those little rectangular areas in which you can type something. The term stems from database design, but it is now used pretty broadly by the design community. Often, when a form is designed, specific fields are marked in some way, usually with an asterisk (*) to indicate that the visitor has to fill something out in order to complete the form—a so-called **required field**. Alternatively, the field may simply represent supplementary information that the site owner would like to collect, but which isn't absolutely necessary in order to complete a transaction. In fact, within the European Union, it is actually illegal to require a visitor to provide this kind of "nice-to-have" data. Curiously, when I recently signed up for a free e-paper on a major U.S. publisher's website, I was required to provide credit-card details! But I digress. . . .

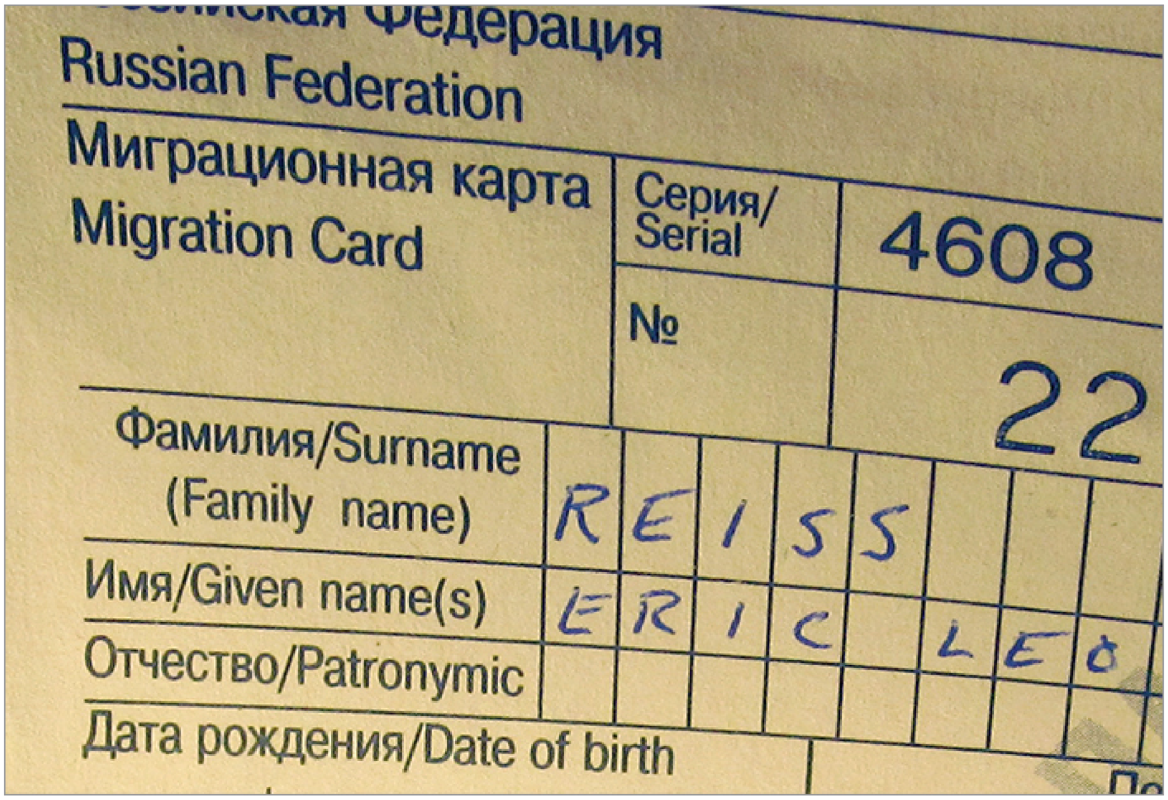
If you are a designer working on a website designed primarily for visitors from the United States, it is tempting to make *State* a required field when folks are asked to provide an address. And if you also are catering to visitors from Canada, you probably want to use more encompassing wording such as *State/Province*.

As it happens, I live in Denmark—imagine an entire country that's no bigger than Houston or Miami in terms of population. Not surprisingly, Denmark doesn't have "states." Actually, most of Europe doesn't have states, provinces, or regions. That means if you make this a required field, there is no way for a large part of the world to complete this form.

This is one of those situations where a programmer or designer may create something that works perfectly well for one group of visitors, but ends in catastrophe for others. Because most of these state/province fields actually have a drop-down list of options, it would certainly help Europeans if "None" was an option. And what about the Australians, who *do* have states, but not the ones on the American lists? One solution to this problem might be to simply ask for the country *before* asking for other address information and have the form adjust itself as needed. (If your programming team thinks this is simply too much bother, then you really need to ask yourself why you're wasting time thinking about usability at all.)

Anyway, when testing a form, make sure folks can reasonably provide all of the information needed to complete the form. I am firmly convinced that this is without question the single greatest reason for conversion failures.

You fill out this entry card before you enter the Russian Federation. But unless you are Russian, the concept of a "patronymic" will probably be something of a mystery to you. Hence, this form is confusing to most foreigners.



Федерация
Russian Federation

Миграционная карта
Migration Card

Серия/Serial 4608

№ 22

Фамилия/Surname (Family name) R E I S S

Имя/Given name(s) E R I C L E O

Отчество/Patronymic

Дата рождения/Date of birth

Forms and business rules

Field validations help the computer make sure it is getting data that it understands and can file appropriately in its database. They are there to check syntax, make sure there are enough numbers in a credit-card number, and so on. The problem is that these rules are invisible to the user, which means that the opportunities for error are enormous.

For example, if you want users to type in a credit-card number, some folks type in spaces between the four-digit segments; others just type in the 16-digit number in one long string. If your validation rules are inflexible, only accepting one of these input options, you're going to frustrate a lot of people. Obviously, the system needs to get 16 digits, not counting spaces. That's a legitimate requirement. But fussing about the spacing is nonsense. It's easy to program things in a more flexible way, and you should make sure someone does so.

Telephone numbers, addresses, postal codes, dates, and all kinds of data (generally numerical) tend to cause problems. Even when I can get a site to accept that I don't have a state, I often find that it still won't accept my four-digit postal code or the Danish spelling of my street name (Strandøre).

When testing the business rules, the idea is not to see what works, but to try and break the system. Ask your family to take a shot at it. This is a remarkably effective way to spot some basic problems.

Interdependent forms

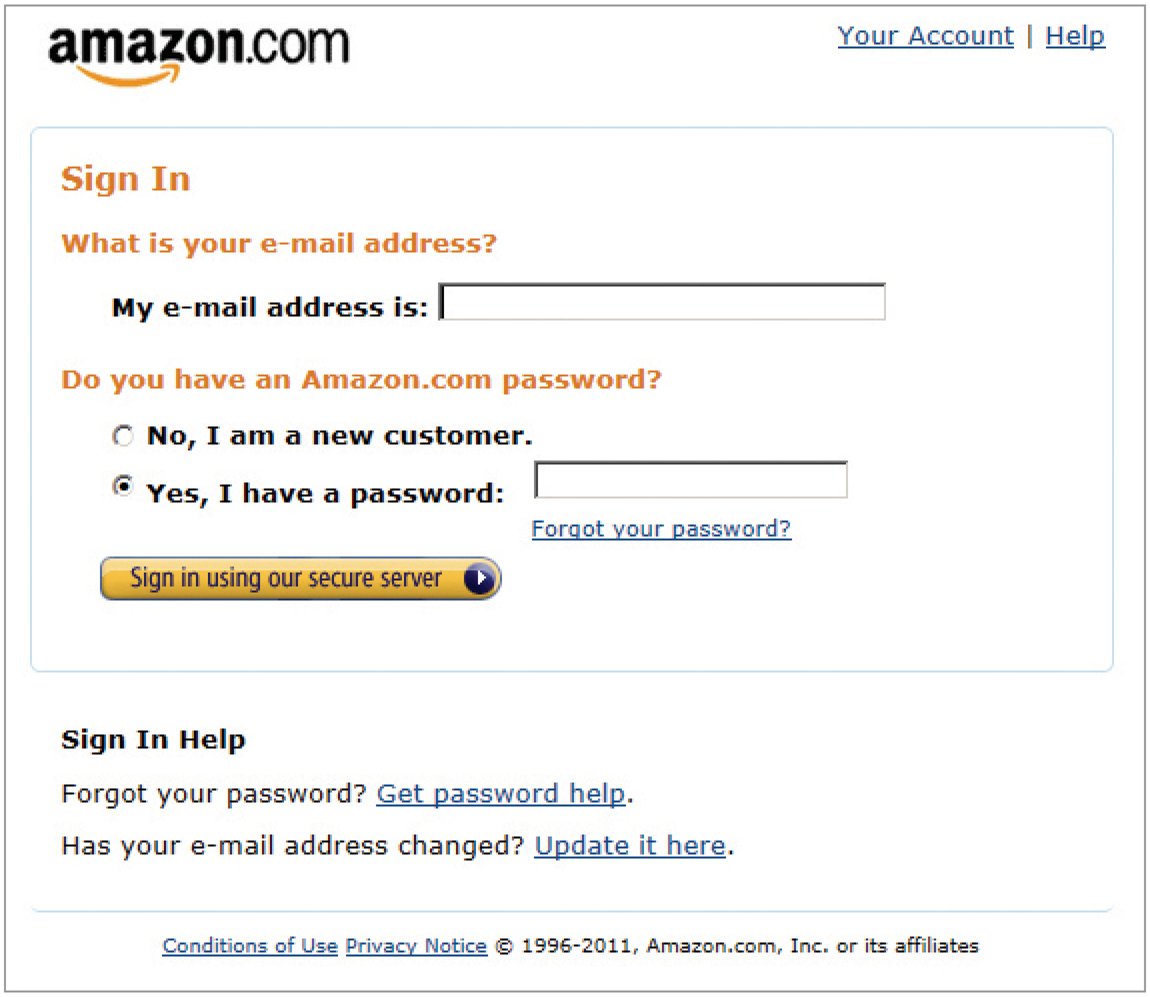
For some reason, online ticketing sites for movie theaters where I live let me choose my seats and get fairly far along in the purchase process before they suddenly ask me to provide user-registration info—a username and password. Honestly, I go to the movies so infrequently that any registration information I may have submitted on a previous visit has long since been forgotten. And if I don't
have it, I must interrupt my task to complete another task that seems more for the benefit of the site owner than for me.

Recently, my wife booked tickets so she could take our granddaughter to *Disney on Ice*. Eventually, she located a ticketing website, found good seats, and was about to pay ,when she was suddenly required to register her personal data with the site owner. Déjà vu! What made this particular situation interesting was that the website gave her just five minutes to complete this task or she would lose the seats and have to start over. Alas, the registration process took almost 10 minutes due to slow servers and other technical limitations. All in all, it took her almost 30 minutes to book the two tickets. She was furious, vowing never again to use this miserable website—and by extension, was mad at the Disney organization that actually had nothing to do with the operation of the ticketing website. (There’s a service-design lesson to be learned here.)

Naturally, some interdependent forms, such as several sequential pages in a shopping cart, are not nearly as odious. The problem occurs when a website breaks a sequential process by asking the user to do something else before continuing on his or her journey. The perception of a user experience is formed as folks move along a path leading from one interaction to the next. Don’t get in the way.

In short, if you have two different forms that need to be completed, make sure folks see these in the appropriate order. And for goodness sake, give folks enough time to fill out both forms satisfactorily before you time them out!

Happily, the login page for Amazon turns up early in the customer journey, so getting through the check-out is easy and straightforward.



Your Account | Help

Sign In

What is your e-mail address?

My e-mail address is:

Do you have an Amazon.com password?

No, I am a new customer.

[✓ Yes, I have a password:]

Forgot your password?

Sign In Help

Forgot your password? Get password help.

Has your e-mail address changed? Update it here.

Conditions of Use Privacy Notice © 1996-2011, Amazon.com, Inc. or its affiliates

Instructions and functionality

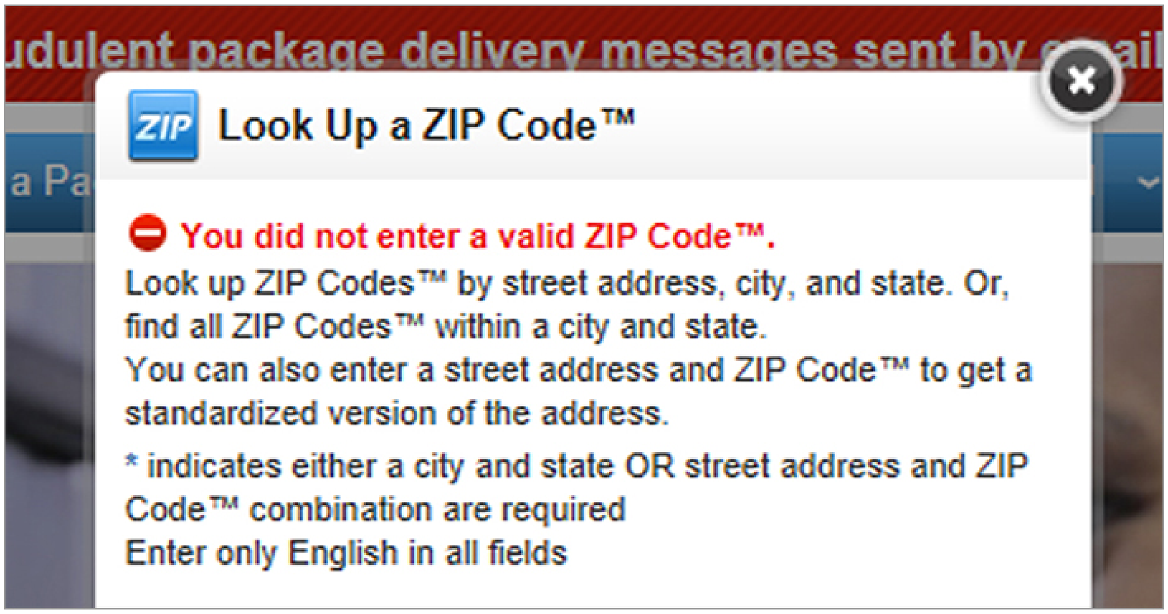
I'm always amazed at websites and other stuff that ask me to do something very specific and then complain when I do exactly what I've been asked to do. This often happens when the person who wrote the instructions (or documentation) has no contact with the designer/programmer and vice versa. Here are two quick examples.

Years ago, I had a crazy VCR made by the German manufacturer Saba. I wish I still had it because it was a classic example of an over-designed machine—there were no fewer than 46 buttons on the front panel! About half of them were labeled in German, and the other half in English. The main power, for example, was indicated in English: Off/On. But the timer function was labeled with the German equivalent: Auf/Zu.

Already here you can see that there is a basic cognitive problem, particularly if the user doesn’t happen to speak German. But to make matters worse, the rather huge instruction book that came with the rather huge machine often reversed things. For example, it insisted I push Zu to turn on the main power and On to activate the timer, which was the exact reverse of the button legends printed on the machine itself. Needless to say, it took a bit of experimentation to get the beast to work.

The lesson is this: When testing stuff, follow whatever instructions you have been given *to the letter*! If the instructions don’t work or don’t make sense, you are going to run into functional problems, so be on the lookout for these kinds of issues and fix them. It’s also a good idea to keep everything in the same language. Think about this the next time you visit an international website that mixes and matches languages on the same page.

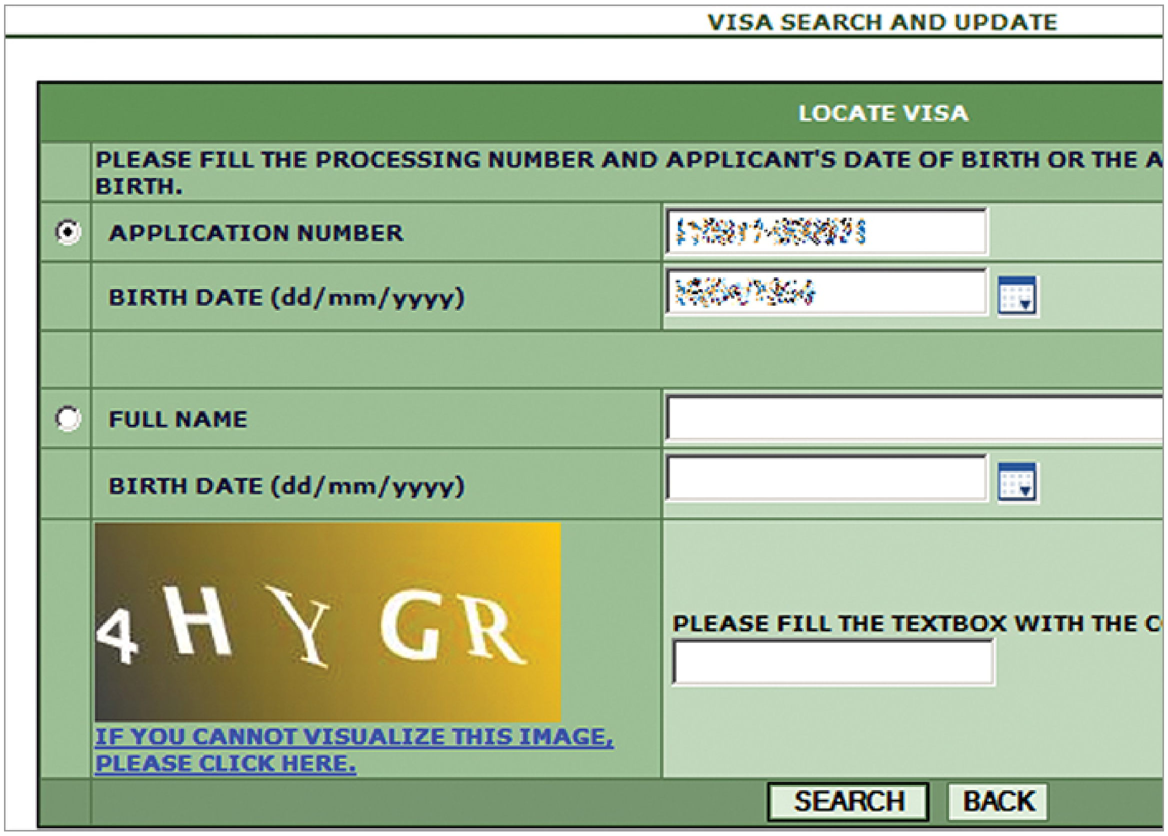
The United States Postal Service has a convenient ZIP Code™ finder. But why did the designers make ZIP Code a required field? This is precisely the information people are looking for!



You did not enter a valid ZIP Code™.
Look up ZIP Codes™ by street address, city, and state. Or, find all ZIP Codes™ within a city and state.
You can also enter a street address and ZIP Code™ to get a standardized version of the address.
* indicates either a city and state OR street address and ZIP Code™ combination are required
Enter only English in all fields

The second example I'd like to share is from a form on the Brazilian Embassy's website that asked me for a date (I was applying for a visa). The instructions in parentheses next to the form field told me specifically to enter my data using the following format: dd/mm/yyyy (with slashes). However, for reasons known only to the backend developers, the date would only be accepted if I typed in: ddmmyyyy (no slashes). It took quite some time to figure out what was going wrong here when the site refused to accept my submission.

Although this website form from the Brazilian Embassy in Copenhagen tells me exactly how I should enter a date, the business rules for this site actually insist on a very different format without slashes: ddmmyyyy. This is as confusing as it is frustrating.


VISA SEARCH AND UPDATE

LOCATE VISA

PLEASE FILL THE PROCESSING NUMBER AND APPLICANT'S DATE OF BIRTH OR THE A
BIRTH.

●	APPLICATION NUMBER	[redacted]
	BIRTH DATE (dd/mm/yyyy)	[redacted]
○	FULL NAME	
	BIRTH DATE (dd/mm/yyyy)	

IF YOU CANNOT VISUALIZE THIS IMAGE, PLEASE CLICK HERE.

PLEASE FILL THE TEXTBOX WITH THE C

Honestly, it is a fairly simple matter to get a database to ignore the slashes, dashes, spaces, and anything else people might type in this field. And to actually *ask* for a particular format and then reject the data is a sure-fire recipe for disaster.

Navigation: Getting folks where they want to go

The second of my three main functionality points deals with the responsiveness of navigation, which is closely related to the third point—processing speed. There are actually two sides to this issue. One side deals with the cognitive feedback from a site or device, which I talk about in the next chapter. The other deals with speed and efficiency, which is what I'd like to talk about now.

My crappy new TV

I recently bought a cheapo LED TV to put in our spare bedroom. It is very shiny and skinny and has a wonderfully sharp picture. But it has the reaction time of an anesthetized turtle. Each time I punch up a new channel, the TV takes five to eight seconds to respond. Needless to say, it is virtually impossible to zap around looking for something interesting to watch. Today, my family does not let me use it unless I have a *TV Guide* in front of me; they are convinced I will die of apoplexy if I don’t have a precise viewing plan *before* I turn the thing on.

But guess what? It turns out I’m not the only impatient person on this planet. When it comes to websites and conversion factors, there is a growing body of proof that the faster a page responds to your request, the better the conversion. Google and Amazon have both documented how cutting response times by as little as half a second can provide major conversion improvements.

One of the really good articles on the subject is by Steve Souders. Even though it was written a while back (2009), it certainly indicates a clear trend. For example, when Shopzilla sped things up from approximately seven seconds to two seconds, they experienced a 25 percent increase in page views, a 7 to 12 percent increase in revenue, and a 50 percent reduction in hardware. Suffice it to say, this is an important issue. Google the title “Velocity and the Bottom Line” if you want all the details.

As to testing any stuff you may have, if you think it seems slow, I guarantee you others will think that it’s even slower. So figure out if there is something you can do to improve the situation. Compressing the file size of photos and graphics is a really good place to start and can be done by anyone with access to a simple graphics program such as Photoshop. (By the way, the rule of thumb is that whatever