About Face 3

The Essentials of Interaction Design

An international bestseller,
now completely revised and updated

Alan Cooper,
Robert Reimann, and David Cronin

About Face 3: The Essentials of Interaction Design
Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published by Wiley Publishing, Inc., Indianapolis, Indiana
Published simultaneously in Canada
ISBN: 978-0-470-08411-3
Manufactured in the United States of America
10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data:
Cooper, Alan, 1952-
About face 3 : the essentials of interaction design / Alan Cooper, Robert Reimann, and Dave Cronin.
p. cm.
Includes bibliographical references.
ISBN 978-0-470-08411-3 (pbk.)
1. User interfaces (Computer systems) 2. Human-computer interaction. I. Reimann, Robert. II. Cronin, Dave, 1972- III. Title. IV. Title: About face three.
QA76.9.U83C6596 2007
005.4'38--dc22
2007004977

3

Beginners, Experts, and Intermediates

Most computer users know all too well that buying a new cell phone or opening the shrink-wrap on a new software product augurs several days of frustration and disappointment spent learning the new interface. On the other hand, many experienced users of a digital product may find themselves continually frustrated because that product always treats them like rank beginners. It seems impossible to find the right balance between catering to the needs of the first-timer and the needs of the expert.

One of the eternal conundrums of interaction and interface design is how to address the needs of both beginning users and expert users with a single, coherent interface. Some programmers and designers choose to abandon this idea completely, choosing instead to segregate the user experiences by creating wizards for beginners and burying critical functionality for experts deep in menus. Of course, no one wants to deal with the extra labor associated with moving through a wizard, but the leap from there to knowing what esoteric command to select from a series of long menus is usually a jump off a rather tall cliff into a shark-infested moat of implementation-model design. What, then, is the answer? The solution to this predicament lies in a different understanding of the way users master new concepts and tasks.

Perpetual Intermediates

Most users are neither beginners nor experts; instead, they are *intermediates*.

The experience level of people performing an activity tends, like most population distributions, to follow the classic statistical bell curve (see Figure 3-1). For almost any activity requiring knowledge or skill, if we graph number of people against skill level, a relatively small number of beginners are on the left side, a few experts are on the right, and the majority — intermediate users — are in the center.

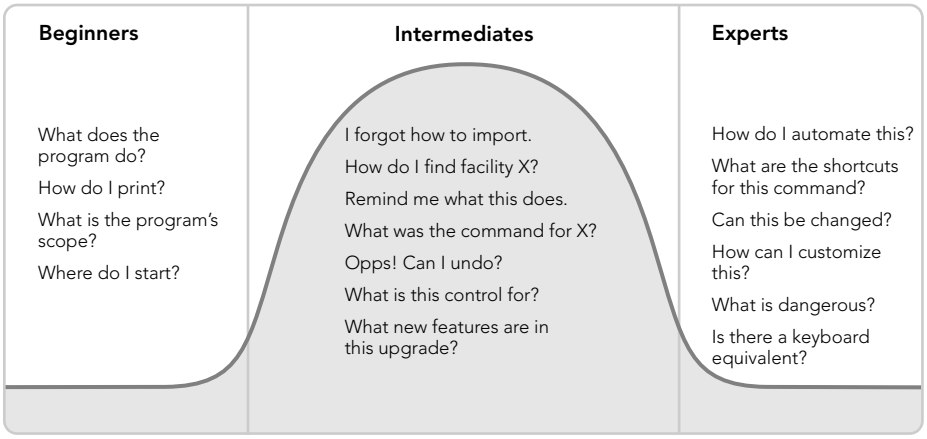

Figure 3-1 The demands that users place on digital products vary considerably with their experience.

Statistics don't tell the whole story, however. The bell curve is a snapshot in time, and although most intermediates tend to stay in that category, the beginners do not remain beginners for very long. The difficulty of maintaining a high level of expertise also means that experts come and go rapidly, but beginners change even more rapidly. Both beginners and experts tend over time to gravitate towards intermediacy.

Although *everybody* spends some minimum time as a beginner, *nobody* remains in that state for long. People don't like to be incompetent, and beginners, by definition, are incompetent. Conversely, learning and improving is rewarding, so beginners become intermediates very quickly — or they drop out altogether. All skiers, for example, spend time as beginners, but those who find they don't rapidly progress beyond more-falling-than-skiing quickly abandon the sport. The rest soon move off of the bunny slopes onto the regular runs. Only a few ever make it onto the double-black diamond runs for experts.

Nobody wants to remain a beginner.

Most occupants of the beginner end of the curve will either migrate into the center bulge of intermediates, or they will drop off of the graph altogether and find some product or activity in which they *can* migrate into intermediacy. Most users thus remain in a perpetual state of adequacy striving for fluency, with their skills ebbing and flowing like the tides depending on how frequently they use the product. Larry Constantine first identified the importance of designing for intermediates, and in his book *Software for Use*, he refers to such users as **improving intermediates**. We prefer the term **perpetual intermediates**, because although beginners quickly improve to become intermediates, they seldom go on to become experts.

Many popular ski resorts have a gentle slope for learning and a few expert runs to really challenge the serious skier. But if the resort wants to stay in business, it will cater to the perpetual intermediate skier, without scaring off the beginner or insulting the expert. The beginner must find it easy to matriculate into the world of intermediacy, and the expert must not find his vertical runs obstructed by aids for trepidatious or conservative intermediates.

In many cases, a well-balanced user interface takes the same approach. It doesn't cater to the beginner or to the expert, but rather devotes the bulk of its efforts to satisfying the perpetual intermediate. At the same time, it provides mechanisms so that both of its smaller constituencies can be effective.

Most users in this middle state would like to learn more about the product but usually don't have the time. Occasionally, the opportunity to do so will surface. Sometimes these intermediates use the product extensively for weeks at a time to complete a big project. During this time, they learn new things about the product. Their knowledge grows beyond its previous boundaries.

Sometimes, however, they do not use the product for months at a time and forget significant portions of what they knew. When they return to the product, they are not beginners, but they will need reminders to jog their memory back to its former state.

In some specialized products, it is appropriate to optimize the user experience for experts. In particular, tools that technically minded people rely on for a significant portion of their professional responsibilities should be inflected towards a high degree of proficiency. Development tools often fall into this category, as do scientific instrumentation and medical devices. We expect the users of those products to come to the table with the necessary technical knowledge, and to be willing to invest significant time and effort to mastering the application.

Similarly, there are other products, especially those used in a transient manner, or those used by people with certain disabilities, that must be optimized for beginners. Examples we have worked on include informational kiosks designed for public spaces like museums, or a device that helps elderly patients with diminished abilities take their blood pressure.

We are often asked whether consumer Web sites should be optimized for beginners or intermediates. Ultimately, we believe that the same considerations that we apply to other digital products should be used here. A well-designed Web site interface should help its user become quickly familiar and comfortable with navigation and functionality. Something worth considering here is that even a customer who has visited your site several times before and may be familiar with what you offer and with Web interaction idioms in general, may not visit your site frequently enough to memorize organizational constructs. This increases the importance of making interactions on your site as transparent and discoverable as possible. Also, as it has become increasingly popular to provide an adaptive experience by tracking user actions on a Web site, it is often useful to rely on cookies to identify a new visitor and to provide unobtrusive orientation assistance for the first few visits to the site.

Designing for Different Experience Levels

Now let's contrast our bell curve of intermediates with the way that software is developed. Programmers qualify as experts in the software they code because they have to explore every possible use case, no matter how obscure and unlikely, to create program code to handle it. Their natural tendency is to design implementation-model software with every possible option given equal emphasis in the interaction, which they, as experts, have no problem understanding.

At the same time, sales, marketing, and management often demonstrate the product to customers, reporters, partners, and investors who are themselves unfamiliar with the product. Because of their constant exposure to beginners, these professionals have a strongly biased view of the user community. Therefore, it comes as no surprise that sales and marketing folks lobby for bending the interface to serve beginners. They demand that training wheels be attached to the product to help out the struggling beginner.

Programmers create interactions suitable only for experts, while the marketers demand interactions suitable only for beginners, but — as we have seen — the largest, most stable, and most important group of users is the intermediate group.

It's amazing to think that the majority of real users are typically ignored, but more often than not that is the case. You can see it in many enterprise and commercial software-based products. The overall design biases them towards expert users, while at the same time, cumbersome tools like wizards and Clippy are grafted on to meet the marketing department's perception of new users. Experts rarely use them, and beginners soon desire to discard these embarrassing reminders of their ignorance. But the perpetual intermediate majority is perpetually stuck with them.

Optimize for intermediates.

Our goal should be neither to pander to beginners nor to rush intermediates into expertise. Our goal is threefold: to rapidly and painlessly get beginners into intermediacy, to avoid putting obstacles in the way of those intermediates who want to become experts, and most of all, to keep perpetual intermediates happy as they stay firmly in the middle of the skill spectrum.

We need to spend more time making our products powerful and easy to use for perpetual intermediate users. We must accommodate beginners and experts, too, but not to the discomfort of the largest segment of users. The remainder of this chapter describes some basic strategies for accomplishing this.

What beginners need

Beginners are undeniably sensitive, and it is easy to demoralize a first-timer, but we must keep in mind that the state of beginnerhood is *never* an objective. Nobody wants to remain a beginner. It is merely a rite of passage everyone must experience. Good software shortens that passage without bringing attention to it.

As an interaction designer, it's best to imagine that users — especially beginners — are simultaneously very intelligent and very busy. They need some instruction, but not very much, and the process has to be rapid and targeted. If a ski instructor begins lecturing on snowpack composition and meteorology, he will lose his students regardless of their aptitude for skiing. Just because a user needs to learn how to operate a product doesn't mean that he needs or wants to learn how it works inside.

Imagine users as very intelligent but very busy.

On the other hand, intelligent people always learn better when they understand cause and effect, so you must give them an understanding of why things work as they do. We use mental models to bridge the contradiction. If the represented model of the interface closely follows the user's mental model (as discussed in Chapter 2), it will provide the understanding the user needs without forcing him to figure out the implementation model.

Getting beginners on board

A new user must grasp the concepts and scope of the product quickly or he will abandon it. Thus, the first order of business of the designer is to ensure that the product adequately reflects the user's mental model of his tasks. He may not recall from use to use exactly which command is needed to act on a particular object, but he will definitely remember the relationships between objects and actions — the important concepts — if the interface's conceptual structure is consistent with his mental model.

To get beginners to a state of intermediacy requires extra help from the program, but this extra help will get in their way as soon as they become intermediates. This means that whatever extra help you provide, it must not be fixed into the interface. It must know how to go away when its services are no longer required.

Standard online help is a poor tool for providing such beginner assistance. We'll talk more about help in Chapter 25, but its primary utility is as a reference, and beginners don't need reference information; they need overview information, such as a guided tour.

A separate guide facility — displayed within a dialog box — is a fine means for communicating overview, scope, and purpose. As the user begins to use the product, a dialog box can appear that states the basic goals and tools of the product, naming the main features. As long as the guide stays focused on beginner issues, like scope and goals, and avoids perpetual intermediate and expert issues (discussed below), it should be adequate for assisting beginners.

Beginners also rely heavily upon menus to learn and execute commands (see Chapter 22 for a detailed discussion about why this is true). Menus may be slow and clunky, but they are also thorough and verbose, so they offer reassurances. The dialog boxes that the menu items launch (if they do so at all) should also be (tersely) explanatory, and come with convenient Cancel buttons.

What experts need

Experts are also a vital group because they have a disproportionate influence on less experienced users. When a prospective buyer considers your product, he will trust the expert's opinion more than an intermediate's. If the expert says, "It's not very good," she may mean "It's not very good for experts." The beginner doesn't know that, however, and will take the expert's advice, even though it may not apply.

Experts might occasionally look for esoteric features, and they might make heavy use of a few of them. However, they will definitely demand faster access to their regular working set of tools, which may be quite large. In other words, experts want shortcuts to everything.

Anyone who uses a digital product for hours a day will very quickly internalize the nuances of its interface. It isn't so much that they *want* to cram frequently used commands into their heads, as much as it is unavoidable. Their frequency of use both justifies and requires the memorization.

Expert users constantly, aggressively seek to learn more and to see more connections between their actions and the product's behavior and representation. Experts appreciate new, powerful features. Their mastery of the product insulates them from becoming disturbed by the added complexity.

What perpetual intermediates need

Perpetual intermediates need access to tools. They don't need scope and purpose explained to them because they already know these things. ToolTips (see Chapter 23) are the perfect perpetual intermediate idiom. ToolTips say nothing about scope and purpose and meaning; they only state function in the briefest of idioms, consuming the least amount of video space in the process.

Perpetual intermediates know how to use reference materials. They are motivated to dig deeper and learn, as long as they don't have to tackle too much at once. This means that online help is a perpetual intermediate tool. They use it by way of the index, so that part of help must be very comprehensive.

Perpetual intermediates establish the functions that they use with regularity and those that they only use rarely. The user may experiment with obscure features, but he will soon identify — probably subconsciously — his frequently used working set. The user will demand that the tools in his working set be placed front and center in the user interface, easy to find and to remember.

Perpetual intermediates usually know that advanced features exist, even though they may not need them or know how to use them. But the knowledge that they are there is reassuring to the perpetual intermediate, convincing him that he made the right choice investing in this product. The average skier may find it inspirational to know that there is a really scary, black-diamond, expert run just beyond those trees, even if she never intends to use it. It gives her something to aspire to and dream about, and it gives her the sense that she's at a good ski resort.

Your product's code must provide for both rank amateurs and all the possible cases an expert might encounter. Don't let this technical requirement influence your design thinking. Yes, you must provide those features for expert users. Yes, you must provide support for beginners. But in most cases, you must apply the bulk of your talents, time, and resources to designing the best interaction possible for your most representative users: the perpetual intermediates.