For a user-friendly interface

Why is it so difficult to get something that is easy to use?

JHUMKEE IYENGAR

HAVE you tried to sell your camera on the Internet and wondered, after your fourth try, whether it's worth it and gave up? Have you tried to book an airline ticket using your frequent flyer miles and after wandering for 25 minutes in vain looking for where the information is buried, just quit? Have you tried to get your three favourite songs to play on your cellphone and after an hour spent on your cell, the operator's website and music sites, just thrown your hands up?

Welcome to life in the 21st century, where software is an intrinsic part of our lives and nothing is simple anymore.

Why does this happen? Because:

○ Users come to a site with particular goals and expectations in mind. To pay a bill, find the lowest fare, search for a press article, buy a book etcetera, and

○ Designers of the software often do not have a clue what those goals are.

Users do not find the information they want, get lost/confused and log off. It's a loss of revenue, goodwill and potential referrals. Yet, you hear the term "easy to use" all the time. Since it is recognised as an important component for customer satisfaction and business success, it is claimed virtually by everyone.

Why, then, is "easy to use" so hard to get? Surely, engineers do not deliberately create unusable designs. Why are things so unfriendly? Because it takes more than engineers to design a success. It takes a Sociology + Engineering + Psychology + Graphic Design perspectives. In typical design processes, software firms ignore all except the engineering perspective. And maybe some of the Graphic Design perspective. They also assume that ease-of-use will automatically happen.

They forget that it takes planning. This involves an in-depth understanding of the engineer's mental map *and* the user's. It is the intersexion space of the two goal-seeking sets that offers the platform for success. But typically, software is created only from a technology perspective, with the user left out. This is what leads to "usage shock", the result of the disconnect between the human mind and the computer mind. The human mind expects one thing, and the computer provides something else.

The goals are different. The engineer's goal is to make the product work and hence ease of development drives the effort. The user's goal is to have needs satisfied. An engineer may claim to be a "user" himself, but being an expert, is only deluding himself.

Today, most firms pay lip service to usability by ensuring a pretty screen for the bosses to marvel at. But a pretty screen is no guarantee of the user experience

The user thinks, *"Oh, I guess it's under 'Special Services'. No, I don't see it here. Okay, how about 'Other'? No not here either'... what the heck does Redemption Qualifiers mean?'*, while the developer thinks, *"Redemption is under Memberships, isn't it obvious?"*

Design must take into account people as they are, not as computers would like them to be. If software must get user-friendly, developers must start talking to sociologists, psychologists and linguists. Psychology principles like attention, memory, perception and so on all have a major role to play. For example, we must display no more than about seven items on a list. And we must not expect users to remember a small chunk of information longer than 20 seconds after viewing. These are the limits of human short-term memory.

Thankfully, ease-of-use can be "managed". It can be planned, predicted, designed and measured. For this, the method of analysis and design must involve actual end users and their actual expectations (not just stated ones). This is a new field called Usability Engineering. It involves first understanding user behaviour with the product in use (say, at a bank, supermarket cash counter or hospital reception). Then, it tests the software in a usability lab, before the final encoding is done.

That contains costs by reducing rework, development time, training time, support costs and others. It maximises usage, productivity, brand appeal and profitability. Some businesses are put off by the fuzzy-wuzzy sound of all this. But mapping the user's mental model is not some eclectic art. Humans expect a system to behave in a certain way, based on their own internal thought processes, and if the product's design matches this, it is easy to use. So, what happens in the mind needs to be understood.

Today, most firms pay lip service to usability by ensuring a pretty screen for the bosses to marvel at. But a pretty screen is no guarantee of the user experience. This depends on the basic design structure, the "brain" behind the screen, which is prohibitively expensive to rework. It is much better to have had the end user in mind at the very onset of the enterprise.

Once we recognise the fact that designing from both sides of the screen makes business sense, we can refine our methods to include both technology *and* the user. Usability Engineering has proven itself with many success stories already. As developers and designers of products, we must not unleash convoluted designs onto hapless victims. Easy does not have to be that difficult!

The author is a consultant with User In Design and Persistent Systems. These are her personal views