

User Experience Research

In 2005 and early 2006, Pfeiffer Consulting conducted an extensive research project collecting information about Macintosh and Windows operating systems. During the research interviews, which included users of both platforms, many Macintosh users stated that they found their computer ³more fluid², more productive, easier to use. They were, however, most often at a loss when they were asked to quantify their perceptions.

These recurring statements were intriguing for us: from a purely functional perspective, both operating systems have become increasingly similar, and even in terms of user interface, the basic concepts and user interface paradigms used by Windows and Macintosh are almost identical.

This discrepancy between user perception and technical features led us to have a closer look at user interface differences, usability, and productivity. During this research, we realized that the terms and concepts we use to analyze technology have remained surprisingly simplistic given the importance digital tools and devices play in our life.

How do we compare technology?

In our innovation-driven society, we tend to compare technology almost exclusively by looking at features and performance. More pixels equals better camera (or so we think), more gigabytes of storage means a better music player, and the list goes on, and on. Of course, in emerging fields of technology, this functionality-centric approach is utterly natural. (Even "cut and paste" was a big deal a long time ago, yet you would hardly find feature it in a product description today.) The problem is that as technology matures, features are not that important any more.

So if we don't look at features any more, what DO we look when we try to decide on the comparative merits of two products? Design? Style? Both are difficult to measure, and don't do much on their own. Of course there is always "user experience", an increasingly important aspect in the success of technology-related products, but an equally elusive one.

Which usability are we talking about?

It's not that we lack research on usability and ergonomics either: a quick search on the web reveals thousands of pages on the subject. Unfortunately, most of the available research focuses on web-usability, and somehow tacitly assumes that what applies to a web-site surely will apply to other products as well.

Not so fast: what is deemed a good web-experience is profoundly different from a good user experience with computer software or on a digital device.

Web usability has a lot to do with creating a flawless "first-time" experience; using a device or computer program, on the other hand, is about fluidity and efficiency in very frequently repeated operations.

And that's only part of the problem: interacting with information<which is generally what using the web is about<is completely different from interacting with a device or a computer operating system.

So if we are interested in understanding the real differences between the Macintosh and Windows, say, or between an iPod and another music players, we are out of luck. We even lack the most basic terminology to describe, let alone quantify perceived qualities and problems. If we can quantify the speed of a computer, or it's productivity, why can't we do the same for the efficiency of a user interface or a device?

So maybe we need to sharpen our perception. Maybe we should add some words and concepts to our tools and terminologies. Maybe we need to accept that there is more to technology than features, and that it is time to look at something else than number of features, millions of pixels or hard-drive capacity.

Introducing: User Interface Friction

These considerations led us to come up with a new concept, that has proven extremely useful in conducting technology analysis. Since in nature it is somewhat similar to the physical notion of friction, we called it User Interface Friction (UIF). We use it to describe and quantify the differences in fluidity and reactivity that exist between different operating systems, between software applications, even between different digital devices (music players, cameras, cell-phones, among others).

We can find examples of User Interface Friction in many places. Did you ever notice how menu behavior can slow you down when you are trying to access a command, like selecting a program from the Start menu in Windows? That's User Interface Friction. Did you ever remark how scrolling through long lists of songs on a MP3 player can be annoying? User Interface Friction again. In fact, any user interface has some degree of friction. Some of it we may not notice, despite that fact that it exists, other examples can be severely annoying.

UIF is the resistance imposed upon a user-guided process through the operating system and the way the user interface reacts. In most cases, it has nothing to do with functionality: we use the term User Interface Friction to define the difference in fluidity and productivity that can be observed when running the same program or procedure on different computer systems, or when trying to achieve the goal on two similar digital devices.

User Interface Friction is inherent in any modern, menu-driven computer system and any device that sports a graphical user interface, and depends on a number of aspects, ranging from the speed at which the system displays a menu or sub-menu, to the efficiency of the mouse. Just like the smoothness of the paper or the ink-flow of a pen can impact the speed of handwriting, User Interface Friction affects practically any procedure where the user interacts through the user

interface with the computer system. UIF could also be compared to the reactivity of a car when one presses the accelerator or the breaks.

(A detailed discussion of User Interface Friction, including user interface efficiency measures and productivity benchmarks of UIF can be found in the report User Interface Friction Research published by Pfeiffer Consulting.)

Subliminal does NOT mean insignificant

One of the most important aspects of User Interface Friction is that, while it may seem hardly noticeable, it shapes our overall user experience and can make the difference between a compelling product and an uninspiring one.

Differences between two devices or programs may appear almost subliminal, but the user who compares both will notice them nonetheless, even if he or she could not explain why. Conversely, the sum of little, seemingly insignificant improvements add up and will make the difference between an excellent device and a lesser one.

Creating a "killer user experience" owes a lot to understanding subtle aspects such as User Interface Friction, and that is why I believe it is a very important notion. In many ways, creating an excellent user interface has become the digital equivalent of first-class manufacturing: we need it as users, and we need to understand what contributes to it if we are developing technology.

Andreas Pfeiffer
(posted March 16, 2006)