

## raphical Users

I know that arguments about graphic user interfaces vs. command line interfaces tend to go nowhere fast, but in this case I found a good explanation of the thinking processes of those who prefer each approach. Rather than argue about which is "best", it might help to understand what is going on in the minds of others, especially for software designers.

The web page with the information is located  
<http://www.lava.net/~dewilson/diane/software/humane.html>

Brief quotes:

What is it about users' needs that cannot be summarized in requirements, or that cannot be analyzed through market research? There are several interrelated problems which cause a failure in communications.

- \* Developers typically don't think like users, even when they use their own software.

- \* We (that means all of us, from planning through development and on to marketing) don't live in the same world as the people who must deal with the problems that we're trying to solve.

- \* Finally, our technology is lacking in the "social graces."

<snip>

Why we can't be end users of our own software

Bruce Tognazzini describes interface styles that are appropriate to different types of users. Software developers tend to be introverted intuitives, people who can see and manipulate abstract structures in their minds, structures which become almost as real as anything in the physical world. All too often, such people are relatively poor at communications because they rely on their understanding of these objects in their minds, objects which aren't visible to others. When introverted intuitives drive to work, they follow a map in their minds, and see very little along the way.

When Tognazzini describes how software developers interact with computers, he states, "They are not dependent on a lot of immediate, real-world data. They should do well with human/computer interfaces that are very sketchy, expecting to user to maintain their own 'reality,' their own maps and models of the internal structure of the program." Software developers do well with command line interfaces.

Many end users of applications are more likely to be extroverts, sensories, or both. Extroverts are better communicators, relying on a rich mixture of words and gestures, and make good use of feedback. Sensories perceive and understand the world directly through their perceptions; in extreme cases, the only things that are real to them are the things that they see, hear, touch, smell, and taste--at the moment that their senses perceive these things. Extroverted sensories see everything along the road as they drive to work.

When Tognazzini describes how these people interact with computers, he states, "[They prefer] interfaces that appear 'real,' that do not depend on the memorization of invisible behaviors, that appeal to the real-world sensations of sight, sound, kinesthesia, and so on.... [They] can be expected to do poorly with interfaces that depend on the user to generate their own maps and models of the internal structure of the program." These people do well with visual interfaces that allow direct manipulation.


So what's the point of all this? Introverts and intuitives make up a minority of the population (about 25% each), but they are the dominant types in software development organizations. To put it bluntly, people who communicate based on symbolic manipulation of mental models are designing interfaces for people who want and need a responsive interface that is rich in information, and that accommodates a wide variety of input and working styles.

<snip>

One last observation needs to be made before we move on to the second problem in understanding our users. Don Norman tells us that users interface with their tasks, not with the computer. This places software in the role of a communications medium. Because machines and users do not share a common language, the software must also translate objects and intentions from one medium to another. Those of us who are introverted, intuitive types are perhaps the best suited for understanding and solving the problems in the user's task domain, yet we are the least suited for the job of creating tools that communicate and translate--it simply doesn't come naturally to us, and we understand too many things that can't be seen.

See the web site for the full article

<http://www.lava.net/~dewilson/diane/software/humane.html>

 Milo - CatGuy

<catguy@lamg.com>