

What Does Usability Mean: Looking Beyond ‘Ease of Use’

by Whitney Quesenbery

The definition of usability is sometimes reduced to "easy to use," but this over-simplifies the problem and provides little guidance for the user interface designer. A more precise definition can be used to understand user requirements, formulate usability goals and decide on the best techniques for usability evaluations. An understanding of the five characteristics of usability – effective, efficient, engaging, error tolerant, easy to learn – helps guide the user-centered design tasks to the goal of usable products.

MEANINGS OF USABILITY

The word "usability" has become a catch-phrase for products that work better for their users, but it is difficult to pin down just what people mean by it. Is ‘usability’

1. a result – software that is usable;
2. a process, also called user-centered design, for creating usable software;
3. a set of techniques, such as contextual observation and usability testing, used to achieve that result; or
4. a philosophy of designing to meet user needs?

These different meanings can be described in four key requirements:

- **Usability means thinking about how and why people use a product.**

Good technical writing, like good interaction design, focuses on user’s goals. The first step in creating a usable product is understanding those goals in the context of the user’s environment, task or work flow, and letting these needs inform the design.

- **Usability means evaluation.**

Usability relies on user-feedback through evaluation rather than simply trusting the experience and expertise of the designer. Unlike conventional software acceptance testing, usability evaluation involves watching real people use a product (or prototype), and using what is learned to improve the product.

- **Usability means more than just "ease of use"**

The 5 Es – efficient, effective, engaging, error tolerant and easy to learn – describe the multi-faceted characteristics of usability. Interfaces are evaluated against the combination of these characteristics which best describe the user’s requirements for success and satisfaction.

- **Usability means user-centered design**

Users are satisfied when an interface is user-centered – when their goals, mental models, tasks and requirements are all met. The combination of analysis, design and evaluation all approached starting from the user’s point of view creates usable products.

DEFINING EASE OF USE

The definition of usability in the ISO 9241 standard is:

"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use"

This definition can be expanded, and made more comprehensive, by including five characteristics which must be met for the users of a product:

- Effective
- Efficient
- Engaging
- Error Tolerant
- Easy to Learn

Effective

Effectiveness is the completeness and accuracy with which users achieve specified goals. It is determined by looking at whether the user’s goals were met successfully and whether all work is correct.

It can sometimes be difficult to separate effectiveness from efficiency, but they are not the same. Efficiency is concerned primarily with how quickly a task can be completed, while effectiveness considers how well the work is done. Not all tasks require efficiency to be the first principle. For example, in interfaces to financial systems (such as banking machines), effective use of the system -- withdrawing the correct amount of money, selecting the right account, making a transfer correctly – are more important than marginal gains in speed. This assumes, of course, that the designer has not created an annoying or over-controlling interface in the name of effectiveness.

The quality of the user assistance built into the interface can have a strong impact on effectiveness. The effectiveness of an interface often relies on the presentation of choices in a way that is clearly understandable to the user. The more informative an interface can be, the better users are able to work in it without problems. Good interface terminology will be in the user's language and appropriate to the task.

Another design strategy to increase effectiveness is to offer redundant navigation, especially for ambiguous situations. Although this may create inefficient paths, it allows the user to work effectively by making more than one choice lead to the correct outcome. This can be especially valuable in interfaces which support infrequent users or those often unfamiliar with the content domain.

Efficient

Efficiency can be described as the speed (with accuracy) in which users can complete the tasks for which they use the product. ISO 9241 defines efficiency as the total resources expended in a task. Efficiency metrics include the number of clicks or keystrokes required or the total 'time on task'

It is important to be sure to define the task from the user's point of view, rather than as a single, granular interaction. For example, a knowledge base which doled out small snippets of information might be very efficient if each retrieval was considered one task, but inefficient when the entire task of learning enough to answer a user's question is considered.

Navigation design elements such as keyboard shortcuts, menus, links and other buttons all have an impact on efficiency. When they are well-designed, with clearly expressed actions, less time and effort are needed for the user to make navigation and action choices..

Making the right choices for efficient use of the software depends on an understanding of the users and how they prefer to work. For example, are they likely to use the interface infrequently or to be habitual users who might learn hidden controls and shortcuts? Do they use the keyboard, mouse or other input devices? For example, keyboard shortcuts can be extremely efficient for proficient users who work with the interface intensively. If they are the primary interaction tool, they can slow down users who are unfamiliar with them, or with the software. Similarly, an interface structured around a set of hierarchical choices which may be the best solution for one-time or infrequent users, might be frustratingly slow as the only way of interacting with a frequently-used program.

Engaging

An interface is engaging if it is pleasant and satisfying to use. The visual design is the most obvious element of this characteristic. The style of the visual presentation, the number, functions and types of graphic images or colors (especially on web sites), and the use of any multimedia elements are all part of a user's immediate reaction. But more subtle aspects of the interface also affect how engaging it is. The design and readability of the text can change a user's relationship to the interface as can the way information is chunked for presentation. Equally important is the style of the interaction which might range from a game-like simulation to a simple menu-command system.

Like all usability characteristics, these qualities must be appropriate to the tasks, users and context. The style of engagement that is satisfying for a repetitive work tool is different than an e-commerce site. Even within the same class of interfaces, different users may have widely divergent needs. What is important is that the design meet the expectations *and* needs of the people who must use the interface.

Error Tolerant

The ultimate goal is a system which has no errors. But, product developers are human, and computer systems far from perfect, so errors may occur. An error tolerant program is designed to prevent errors caused by the user's interaction, and to help the user in recovering from any errors that do occur.

Note that a highly usable interface might treat error messages as part of the interface, including not only a clear description of the problem, but also direct links to choices for a path to correct the problem. Errors might also occur because the designer did not predict the full range of ways that a user might interact with the program. For example, if a required element is missing simply presenting a way to fill in that data can make an error message look more like a wizard. If a choice is not made, it can be presented without any punitive language. (However, it is important to note that it is possible for an interface to become intrusive, or too actively predictive.)

For those errors which are out of the control of the interface – system failures or other disasters - take a lesson from flight attendants and quietly, calmly guide the user through the process of helping the program recover from the problem.

Some guidelines for preventing errors are:

- Make it difficult to take incorrect actions. Design links and buttons to be distinctive, use clear language, avoiding technical jargon, and be sure that dependent fields or choices appear together.
- Make it difficult to take invalid actions. Limit choices when possible to those which are correct, provide clear examples for data entry, present only appropriate navigation options.
- Make it difficult to take irreversible actions. Provide the ability to back track, provide means to undo or reverse actions, avoid dead-end screens. Don't indiscriminately use confirmations – users become insensitive to them.
- Plan for the unexpected. Allow for users to add new entries, take exceptional routes through the interface or make choices you did not predict. Be polite about "correcting" mistakes that may arise from this lack of foresight.

Easy to Learn

One of the biggest objections to "usability" comes from people who fear that it will be used to create products with a low barrier to entry, but which are not powerful enough for long, sustained use.

But learning goes on for the life of the use of a product. Users may require access to new functionality, expand their scope of work,

explore new options or change their own workflow or process. These changes might be instigated by external changes in the environment, or might be the result of exploration within the interface.

An interface which is easy to learn allows users to build on their knowledge without deliberate effort. This goes beyond a general helpfulness to include built-in instruction for difficult or advanced tasks, access to just-in-time training elements, connections to domain knowledge bases which are critical to effective use.

Allow users to build on not only their prior knowledge of computer systems, but also any interaction patterns they have learned through use in a predictable way. Predictability is complementary to interface consistency. A consistent interface ensures that terminology does not change, that design elements and controls are placed in familiar locations and that similar functions behave similarly. Predictability expands this to place information or controls where the user expects it to be. This concept has been discussed in connection with Palm Pilot design– and especially important if you make an interface which goes beyond the boundaries of simple platform design standards. Good use of predictability requires careful user analysis and observation, but can make new functions easy to learn by providing controls where the user expects them to be.

WORKING WITH THE FIVE E'S

Finding the right balance between the usability characteristics for the specific design context is an important part of the user analysis. The difference in emphasis is helpful in understanding distinctions between user groups and in thinking through the implications for the interface design. Two fictional examples show this at work.

A Corporate Human Resources (HR) Site

A typical web knowledge management system is used by employees to look up information about their benefits, including options for leave, medical benefits and scholarship support. These users might express the following needs (in order of importance)

- *Effective* – Users were most concerned that they had accurately found all of the options which applied to them, and that they understood all the implications of any choice they made.
- *Easy to Learn* – The site used infrequently. When they did visit it, users needed information about difficult life events, often under personal stress. Users did not expect to gain any mastery of the site, and wanted guidance through any procedures.
- *Efficient* – The previous HR system involved completing paper forms and waiting for an appointment with a specialist – a process that often took several days. Users wanted to get answers more quickly than that. They were willing to spend a reasonable amount of time on the site when it produced answers. They were willing to tolerate minor delays while forms were processed when they got results within minutes.
- *Engaging* – Users wanted a pleasant experience, but were most concerned with a presentation of the material they could understand easily than with "whiz-bang" features

- *Error Tolerant* – They assumed that they could trust the site to make calculations correctly. This characteristic was last in their priority, assuming that the system would not make mistakes.

A Conference Registration System

Contrast the previous example with users of an online conference registration system. These users (also fictional) will use this site once, but are spending a relatively large sum to register. Their experience of the conference itself may depend on the success of the registration system.

- *Efficient* – The users saw registration as a simple task and were not willing to spend much time on it, especially compared to filling in a paper form.
- *Error Tolerant* – They were concerned that the system might make mistakes in processing their choices, and wanted good validation, confirmation and error notification during the process. They also wanted to be sure that they could change their minds without needing to start the process over.
- *Engaging* – Some users expected to have options or features explained during the registration process. All wanted clear, understandable presentation, citing difficult paper and online forms they had encountered in the past as problems.
- *Effective* – They assumed that they would be registered correctly. This characteristic is placed lower on the list because of user emphasis on error handling to prevent problems.
- *Easy to Learn* – Because they saw the task as simple, users assumed that they would be able to complete it without assistance.

Thinking through user's perspective

Although the examples above are fictional, they illustrate one way to use the five usability characteristics to understand the user requirements and mental model for a task. By breaking down the generalized concept of usability into specific areas, the users can be understood in a multi-dimensional way, and usability becomes more than a simple requirement that the program be "easy to use."

A useful exercise is to write a statement for each characteristic for each user group. These statements can be written in the third person (as above) or can be turned into first person statements as a way of capturing a sense of the emotion or tone surrounding each statement. Where direct quotes from users are available, they add richness and credibility. Sometimes the directness of the quote or the diversity of users that the quotes show can be helpful in making users come alive for both to both designers and developers.

There are several benefits of this exercise. The first is to help specify the user groups. When a group of statements seems correct for one user, but not for another, this may be exposing important differences in user requirements. Another is to force the user analyst into a clear and concise expression of user needs. Finally, it can be a useful tool to build a consensus within a team on the user analysis.

This exercise can be done at the beginning of a project, even before any user analysis or observation has been done. In this version, the

work focuses on the group's current understanding of users. Points of disagreement indicate a need for better understanding of users. Points of agreement can be confirmed through analysis. The set of statements for each identified target user group serves as a benchmark for future work.

After user analysis, the exercise is repeated. Places where the team's initial version differs significantly from the post-analysis version need careful attention to be sure the implications for the design are understood.

Connection to usability goals

Usability goals can also be tied to the five characteristics. Each user need statement can be turned into a usability goal or requirements. For example, requirements can be specified with a range of acceptable values, such as:

- Efficient - "The user will be able to successfully complete the registration in under 3 minutes"
- Effective - "Less than 5% of the registrations will have errors, omissions or inconsistencies requiring a follow-up contact by the staff."
- Engaging - "At least 80% of employees will express comfort with using the online system rather than visiting the HR office."
- Error Tolerant – "The system will validate all housing, meal and tutorial choices and allow the user to confirm pricing for these options before completing the registration."
- Easy to Learn – "Users will be able to successfully complete a benefits calculation without needing any external instruction or help screens."

One aspect of transforming archetypal user statements into usability goals must be stressed. Users often place a low importance on characteristics which they simply expect to be well represented in the interface. An example of this is the assumption by the conference registration system user that the task was simple enough that ease of learning was not a critical factor. In creating usability goals, the emphasis must be reversed, with a priority placed on meeting those base-line assumptions. An interface that fails in this will not be usable, even if it meets other requirements. In fact, this basic failure will likely cause failures in other areas. For example, if the registration is difficult to learn, users are likely to take longer to complete the task, exceeding efficiency targets, and be less accurate, failing in effectiveness.

Planning usability evaluations

Understanding specific targets for these usability goals also helps plan usability evaluation. The testing techniques selected may vary, depending on which of the characteristics you are most interested in. Some can be tested with early prototypes or even paper mockups, but others require working software or very high fidelity prototypes.

Characteristic	Type of Usability Evaluation
Efficient	Time (or count clicks or page views) realistic tasks. Must use working versions of the software and plausible sample data.
Effective	Evaluate tasks for how accurately they were completed, and how often they produce errors.
Engaging	User satisfaction surveys or qualitative interviews can gauge user acceptance and attitudes towards the software.
Error Tolerant	Include task scenarios with potential problems in test use scenarios
Easy to Learn	Control how much instruction is given to test participants, or carefully recruit users with different levels of domain knowledge and experience.

In planning usability evaluations, be sure that the most important characteristics are included, and tested in a realistic way. For example, if efficiency is the most important characteristic,

CONCLUSION

Usability and user-centered design are iterative. The work proceeds in a cycle of hypothesis and evaluation, with a picture of users and design solutions to meet their needs building in richness and completeness with each iteration. The five E's (effective, efficient, engaging, error tolerant, easy to learn) provide the practitioner with a set of characteristics which can be used to organize and analyze information from users. They offer trace-ability from initial information-gathering through requirements setting and finally in evaluation. This might allow the understanding of the specific needs around each characteristic to grow, or be an opportunity to confirm whether the user requirements were chosen correctly in the early stages of the project. In either case, they let you go beyond "ease of use" in a practical way and help make it easier to make products more usable.

REFERENCES

- Bergman, E. and Haitani, R. "Designing the Palm Pilot" in Bergman, E. Information Appliances and Beyond. Morgan Kaufmann: San Francisco, 2000
 - ISO 9241-11: 1998 Ergonomic requirements for office work with visual display terminals – Part 11: Guidance on usability
 - Landauer, T. K. The Trouble with Computers: Usefulness, Usability, and Productivity. The MIT Press: Cambridge, MA, 1995.
 - Quesenbery, W. "UPA 99 Workshop Report: Crossing the Chasm - Promoting Usability in the Software Development Community." UPA Common Ground, Vol 10 No 1, March 2000
-

This paper was published in the Proceedings of the 48th Annual Conference, Society for Technical Communication, 2001

The URL for this article is: <http://www.wqusability.com/articles/more-than-ease-of-use.html>

Whitney Quesenbery is a user interface designer and usability specialist with a passion for clear communication. She is president of UPA and was appointed to a committee for the US Election Assistance Commission, where she works to ensure the usability of voting systems. Before she was seduced by a little beige computer, Whitney was a theatrical lighting designer. The lessons from the theatre stay with her in creating user experiences. She can be reached at www.WQusability.com