

## Balancing the 5Es: Usability

by Whitney Quesenbery

Just what do we mean by usability? Before we can set out to achieve it, we need to understand what it is we are trying to achieve. It's not enough to declare that from here on, our software will be more user friendly or that we will now be customer focused.

Functional requirements answer the question, "What does this program have to do?" Usability requirements answer different questions: How do users approach this work? How do they think about the tasks? How do they judge a successful experience?

Some of the confusion is caused by the elasticity of language. It can be hard to find a single word to cover a large concept, and usability is a large concept. The truth is that the word "usability" has become a catch-phrase for a set of ideas about the relationships between users, designers, developers, and the software. In any specific sentence, it might be used to mean:

- A quality of the final product
- A process for creating usable software
- The specific techniques used to achieve that result
- A philosophy of designing with people in mind

I'm going to use the word "usability" as the quality or characteristic of a product (that is, software, Web application, or any other development project you create) that meets the needs of the people who use it, allowing them to work — or play — with it for their own purposes and in a way that is appropriate for them.

Although there are as many methodologies and variations in usability as in any other relatively new practice, there are some common threads:

**A focus on understanding the entire context of use.** It is not enough for someone to be able to navigate through the software; they must be able to complete their tasks, meet their goals, and do so in a way that makes sense in their environment.

**Evaluation and iteration as part of the process.** Most practitioners rely on user feedback through research or evaluation rather than simply trusting the experience and expertise of the designer to get it right the first time. This also means using what you learn to improve the product before it is released.

**A user-centered approach to design.** When product development has been focused on understanding users' goals, mental

**The formal definition of usability from the ISO 9241-11 standards 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" [1].**

models, tasks, and requirements, the entire process shifts from "making it work" to "making it work right." As Kim Goodwin points out in the next article, usability testing and iterative refinement can take an existing product and "make it right." User-centered design takes this one step further — it can help you make the "right product."<sup>1</sup>

**Designing for specific audiences.** Usability means more than just "ease of use." People and software are both complex, and usability is just as multifaceted. It requires a balance of many different aspects of user interaction.

So aren't there any simple standards for usability? Why can't we

<sup>1</sup>See Kim Goodwin, "Are You Making the Product Right or Making the Right Product?" pp. 12-15 in this issue.

just write design and development guidelines and create style guides that will ensure that all applications are usable?

Part of the answer lies in the nature of design. While there are some basic, general principles that we use in design, there are few hard and fast rules. For example, the book *Universal Principles of Design* [2] is a collection of illustrated references for an eclectic group of 100 basic principles, such as the golden ratio, Ockham's razor, alignment, and highlighting. Each is a useful tool for making design decisions, but the book has few recipes. What is universal is the *principle*, not its specific application in a design.

Similarly, usability principles give us a starting point for our work. Usability specialists sometimes joke that the motto of usability is "it depends." What *it* depends on is the need to understand not just good design, or people in general, but the specific people and their specific goals and tasks. The point of all of the work is to allow you to make design decisions with enough information to understand what usability means ... in this context.

## DIMENSIONS OF USABILITY

When I examined the usability literature, I found a number of good lists of qualities of usable software. They included words such as user friendly, memorable, pleasure, accessible, learnable, findability, quality, useful, and error averse. In *Usability Engineering*, Jakob

Nielsen suggests five qualities of a usable product: learnability, efficiency, memorability, errors (low rate, easy to recover), and satisfaction [3].

For my own list, I decided on:

- Effective
- Efficient
- Engaging
- Error tolerant
- Easy to learn

At first, using words that all started with "E" was just a word game.

But I was also looking for a way to make the dimensions of usability easy to remember, and thus the 5Es were born.

### Effective

Effective is the first E. It addresses whether the software is useful and helps users achieve their goals accurately. If users cannot actually do the thing they set out to do (or do something unnecessary), it probably doesn't matter whether the experience is short or long, easy or hard. In the end, they have failed to complete their tasks or meet their goals. If we want to be able to measure effectiveness, we have to understand how people define success or usefulness and whether this is relatively straightforward or more subtle.

### Efficient

Efficiency is the speed (with accuracy) with which work can be done. Efficiency may be something that is carefully defined; for example, in a call center where operators are measured on the number

of calls they can handle in a day. Or it may be a subjective judgment of when a task is taking "too long" or requires "too many clicks."

### Engaging

A simple definition of engaging is how pleasant, satisfying, or interesting an interface is to use. All software has an emotional impact on users, though the importance of this dimension varies with the type of program. In a work application, an engaging interface might draw someone into the work, help the person work with confidence, or present information in a way that is particularly easy to read. The visual presentation and the style or quality of the interaction contribute to making software engaging or off-putting.

### Error Tolerant

Error tolerance involves how well the product prevents errors and helps users recover from any errors that do occur. It would be lovely to say "error free" or "prevents errors," but mistakes and accidents and misunderstandings will happen. The cat nudges the mouse as you click. You misread a link and need to find your way back, or you enter a number with a typo. The real test is how helpful the software is when an error does occur.

### Easy to Learn

Ease of learning concerns how well the product supports both initial orientation and deeper learning. A product may be used just once, once in a while, or on a daily basis. It may support a task that is easy or complex, and the user may be an expert or a novice in this task. But

every time it is used the interface must be remembered or relearned, and new areas of the product may be explored over time.

### A QUESTION OF BALANCE

It would be convenient if each of the dimensions of usability was equally important in every product and for every user, as shown in Figure 1. They are not, however, and this provides one of the first opportunities to work with the 5Es to better understand the usability requirements for a product.

The balance among the 5Es can set the direction for the interface design. Understanding these aspects of usability, in other words, is the beginning of understanding what usability “depends” on.

In Table 1, we can see that two users of a benefits management application — an employee and a benefits specialist — have very different needs. For both users, effectiveness is necessary, but the

needs of the benefits specialist are heavily weighted toward efficiency. For the employee, efficiency takes a backseat to ease of learning, error tolerance, and how engaging the software is.

The value of thinking through the usability needs in this granular way goes beyond the benefits of simply “understanding users.” It can be a tool for project management, helping determine the techniques for both user research and usability evaluation used during the project. It will suggest design approaches and identify places where tradeoffs can be made when necessary. This value extends through every step of the design and development process.

### A USER-CENTERED APPROACH

Most UCD processes follow the general outline of ISO 13407: *Human-Centred Design Processes for Interactive Systems*. They typically start with a discovery

process and then loop through research-design/prototype-evaluate steps until the project is complete. A design approach is created for the whole project, and it is tested to make sure that the structure and organization of the tasks are correct. Then each function is designed, using the same iteration of analysis, design, and evaluation, until all the functions have been integrated into the overall design architecture. Final usability tests provide a last check on the results before the software is released. These processes span the entire lifecycle, bringing usability and interaction design work into all stages of the process, from initial product conception through the entire development period (see Figure 2).

Of course, the intensity of the UCD work varies — it’s high in the design stages and low during implementation. At each stage, the team must decide what UCD activities will best support the product by filling gaps in understanding and answering questions that will impact how the product is designed and built.

### Get Assumptions on the Table

We do not start each new project with a blank slate. Whether the project is a new version of an existing program or a completely new product, the people on the team usually have a history of work in the industry or business domain. They have had successes as well as problems (or even outright failures), and they bring assumptions and beliefs based on that past experience into the new project. Get this

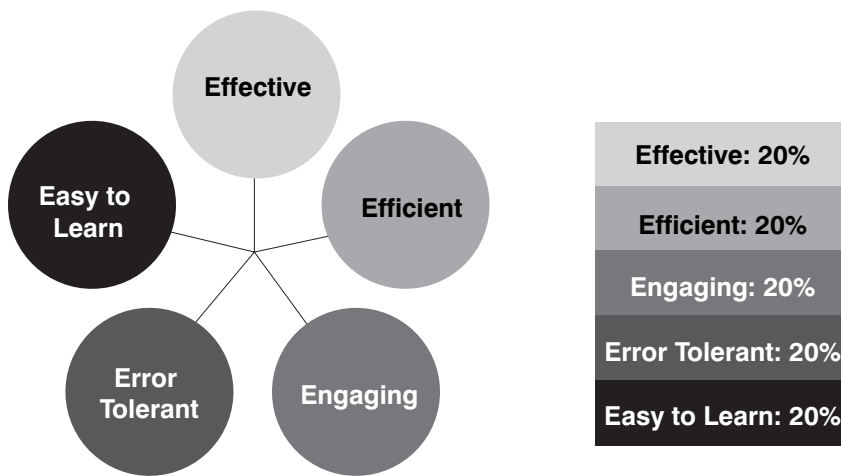


Figure 1 — In this case, the 5Es are evenly balanced.

information out on the table and build a picture of the user and usability requirements as the team envisions them. This is the baseline against which new information from user research can be measured: does it reinforce or contradict the popular wisdom? If a better

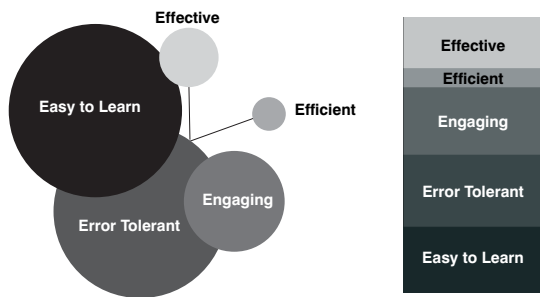
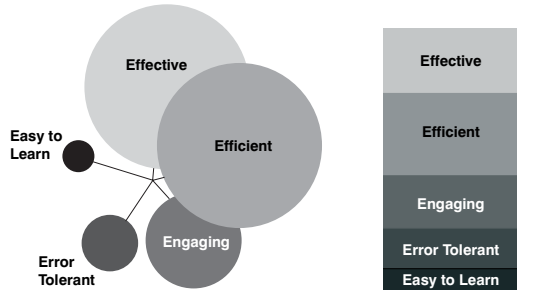
understanding of users and context brings a new picture, this may change design assumptions and even development approaches.

**Learn about Users, from Users**

There are many techniques for learning about users, each of which

can help you discover different aspects of the user experience. If you are concerned about efficiency, you will want to use a technique that lets you see real people completing real tasks in their own environment; for error tolerance, you may want to use critical event

Table 1 — Different Users, Different Usability Needs

<p><b>Employees</b></p> <p>All of the employees of companies that use our benefits management application must use it to make any changes in their personal information or choices of benefits. They use this application infrequently. Before this application was deployed, they usually made these changes by visiting the HR department and filling in a paper form with the help of a benefits specialist. They are often unsure of their options and nervous about doing something that might “mess up” their insurance.</p> <p>They need:</p> <ul style="list-style-type: none"> <li>• Good instructions to replace the personal interview (ease of learning)</li> <li>• Confirmation of not only their data updates, but any impact these changes will have on their benefits (error tolerant)</li> <li>• Reassurance throughout the process and confidence that they were accurate in their entry (engaging/effective)</li> </ul> 	<p><b>Benefits Call Center Specialists</b></p> <p>The company has a call center where benefits specialists can assist employees with any problems or questions. They also complete some processes that cannot be done by the employees themselves. They work in the application every day, often answering the same questions over and over and using the same screens for most of the calls. They have been trained not only in the benefits offered, but also in the use of the application.</p> <p>They need:</p> <ul style="list-style-type: none"> <li>• To be able to complete routine functions quickly (efficient)</li> <li>• A good overview of the employee they are working with on a single screen, so they can focus on the conversation rather than the interface (engaging/efficient)</li> <li>• An way to confirm all changes with the employee before they are made (effective)</li> </ul> 
--	--

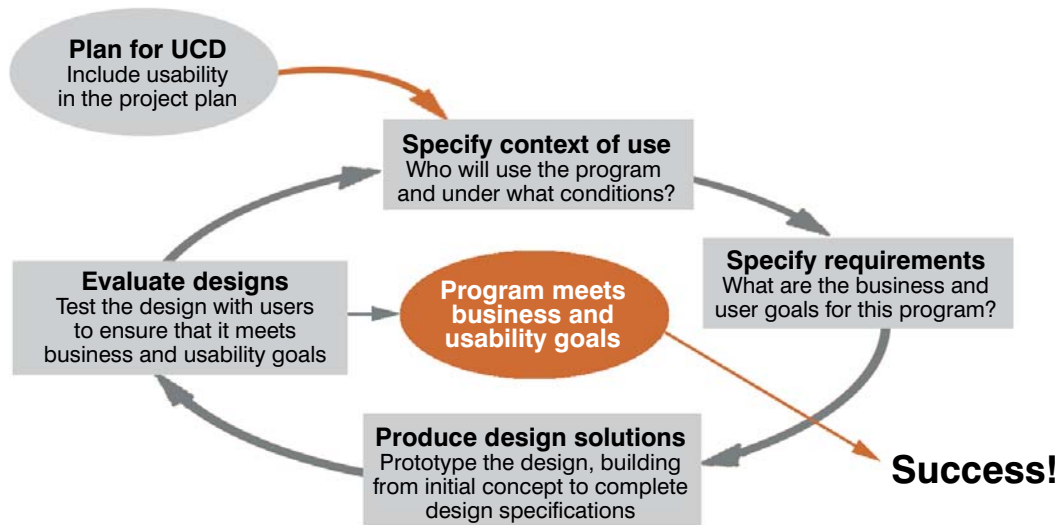


Figure 2 — The user-centered design lifecycle.

analysis and compare that to actual errors reported or logged. Using the different aspects of usability as a tool for selecting research techniques can help ensure that you get the answers to your questions and information that will help you make good design choices.

When the user research and analysis are complete, you have an opportunity to compare your new understanding of users with the team's initial view. This is a chance to update the picture and correct any assumptions that have proved to be incorrect.

The usability literature is full of examples of product innovations

**The usability literature is full of examples of product innovations that were spurred by a changed image of the users.**

that were spurred by a changed image of the users. For example, I once worked on a payroll and employee-management program for small businesses. As we began work on the design, we were told that the typical users worked with many different programs, were familiar with Microsoft Office programs, used e-mail regularly, and were eager to learn to use the program to improve their business. The product development team suggested that the users' most pressing usability need was efficiency, so they could process their payroll quickly; effectiveness, or accuracy, was the second most important need.

When we began working with users, we quickly discovered how inaccurate this portrait was. These small business office managers worked with just one or two programs, usually software specifically designed for their industry — and they thought of this as “a lot of different tools.” They rarely used

general office software from Microsoft or any other vendor, and they did not use e-mail at work (although many had personal accounts). Most importantly, they were interested in learning just enough to get by; they wanted to get their payroll done, not change the way they did business.

In this project, we started our user research focusing on speed and accuracy and trying to learn how users completed the specific task of creating their payroll, but we found that this was the wrong approach. Instead, we changed our techniques to focus on ease of learning and error tolerance. We wanted to know, for example, what aspects of the process the software needed to teach and what kinds of problems most often led to mistakes in the payroll.

### Creating Usability Goals and Requirements

Each of the 5Es can be the basis for a usability goal. A user

Table 2 — The 5Es and Possible Design Approaches

Dimension	User Needs	Possible Design Approaches
Effective	Accuracy	<ul style="list-style-type: none"> <li>• Provide feedback on all critical actions</li> <li>• Eliminate opportunities for error</li> <li>• Provide sufficient information for user decisions</li> </ul>
Efficient	Operational speed	<ul style="list-style-type: none"> <li>• Design navigation for ideal and alternate workflows</li> <li>• Provide shortcuts</li> <li>• Use interaction styles and design widgets that support speed</li> <li>• Minimize extraneous elements on the screen</li> </ul>
Engaging	To be drawn in	<ul style="list-style-type: none"> <li>• Use clear language and appropriate terminology</li> <li>• Set a helpful tone, with a level of conversation suitable for the users</li> <li>• Structure functions to match users' tasks</li> </ul>
Error tolerant	Validation and confirmation	<ul style="list-style-type: none"> <li>• Transform "errors" into alternate paths</li> <li>• Use controls that aid in accurate selection</li> <li>• Be sure actions are easily reversible</li> </ul>
Easy to learn	Just-in-time information	<ul style="list-style-type: none"> <li>• Make the interface helpful with minimalist prompts and instructions provided where they are needed</li> <li>• Create "guided" interfaces for difficult or infrequent tasks</li> </ul>

statement like "How do I know whether everyone will receive the correct bonus on their next pay-check?" might lead to a requirement that the user be able to see and confirm all choices before taking a final action. Or a program with many infrequently used tasks might have a usability goal that it be possible for a (typical, trained) user to complete such tasks without additional training or the use of an external manual.

Whether the statement leads to a functional requirement or a usability goal, tying each of them to one of the usability dimensions connects the statement to that initial conversation and the shared

vision that emerged from it. This can also help reveal any differences in needs, or emphasis, among different users. For example, a manager may care that the work is done efficiently and see it as a "time on task" problem, while workers may see it as a problem of error tolerance and how well the application supports them as they work.

### Forming a Design Approach

A focus on the wrong aspects of usability is a frequent cause of unusable products. Therefore, the design approach should always start from and be tailored to usability requirements. For example, do some users need shortcuts or ways

to handle more than one database record at a time? Or do infrequent users need built-in assistance to "remind" them how to use the interface? Each of the 5Es suggests some possible design requirements (see Table 2).

### Planning Usability Testing

What kind of usability evaluation is needed to ensure that the design has met usability goals? What kinds of prototypes are needed to get useful results? As with user research, the answers depend on the dimensions you are most concerned about (see Table 3). For example, an application that needs to support very efficient operation

Table 3 — The 5Es and Possible Evaluation Techniques

Dimension	Possible Evaluation Techniques
Effective	<ul style="list-style-type: none"> <li>• Create scenarios with difficult or ambiguous tasks.</li> <li>• Evaluate tasks in terms of how successfully they are completed and how often they produce undetected errors.</li> </ul>
Efficient	<ul style="list-style-type: none"> <li>• Construct the test with enough repetitions of typical tasks to create a realistic work rhythm.</li> <li>• Use working software or a high-fidelity prototype.</li> <li>• Observe users at work, looking for situations that interrupt them or slow them down.</li> <li>• Collect timing data, but also interview participants for their subjective impression of the program.</li> </ul>
Engaging	<ul style="list-style-type: none"> <li>• Use satisfaction interview questions or surveys as part of the evaluation.</li> <li>• Do comparative preference testing of presentation design.</li> <li>• Construct the test so that participants are able to abandon a task if they want.</li> </ul>
Error tolerant	<ul style="list-style-type: none"> <li>• Construct scenarios to create situations in which errors or other problems are likely.</li> <li>• Observe how easily or accurately users are able to recover from problems when they occur.</li> </ul>
Easy to learn	<ul style="list-style-type: none"> <li>• Control how much instruction is given to test participants, or recruit participants with different levels of experience or knowledge.</li> <li>• Mix frequently used tasks with functions used less often or tasks with unusual variations.</li> </ul>

probably needs to be tested with a high-fidelity prototype or an early version of the program, with some initial training and a realistic set of tasks matching typical working conditions. To test how well a product engages users in a complex task, however, working with early conceptual prototypes will help focus on the overall process rather than the specific details.

### FITTING USABILITY INTO THE SCHEDULE

One of the most frequent objections to usability or user-centered

design is a practical issue. How can all this extra work fit into schedules that are already crammed to the breaking point? Let's flip this question around and ask a different one: how can usability help cure what ails your software development process?

It's not as though the process of creating software has been pain free. In fact, when you ask developers what they hate the most about their job, they'll tell you:

- Requirements that change, and change, and change

- Customers or business analysts who don't understand what you can — and can't — do with software
- Building something, only to be told that it's not really what users (or marketing) wanted after all

The funny thing is that these are all part and parcel of what the Standish Group said in its *CHAOS Report* back in 1994: 34% of projects are cancelled outright before they are completed, 50% are an impaired version of the original vision, and 16% succeed [4]. And the ones that fail have their source

of failure at the very beginning of the project:

- Lack of user input (12.8%)
- Incomplete requirements and specs (12.3%)
- Changing requirements and specs (11.8%)

As these findings show, many changes in requirements come about because the requirements were not well understood in the first place. Thus, failure to do up-front user research contributes to flawed and unmanageable projects.

Interestingly, these are the very problems that usability professionals and user interface designers complain about:

- Software that doesn't take into account the users' real work, tasks, and environment
- Too much emphasis on technical requirements and not enough balancing user requirements
- Design and usability "added" to the product long after they could have any real influence

In other words, managers, developers, and designers have all identified similar "points of pain." This suggests not only that we see the problems more similarly than we think we do, but that they all spring from the same source. We need a common language for both functional and user requirements, as well as usability evaluation for feedback on our work throughout the development process. Without this common ground, we can

neither describe the product accurately nor recognize when it has been successfully created.

## CONCLUSION

If you want to create a usable and useful product, knowledge and understanding of the people who will use it must be built into the concept and architecture. To quote a favorite usability saying, "Usability is not something that can be spread on like peanut butter at the end of the project."<sup>2</sup>

In today's chaotic world of software development, it is easy to reject new ideas; they add complexity and may seem to reduce control over the development process. However, if it's integrated well, user-centered design can not only help create better products, but also reduce risks and rework. When the product design is informed by an understanding of user needs, it has a much greater chance of meeting those needs.

## REFERENCES

1. ISO/IEC. *9241-11 Ergonomic Requirements for Office Work with Visual Display Terminals (VDT)s – Part 11 Guidance on Usability*. 1998: ISO/IEC 9241-11: 1998 (E).
2. Lidwell, W., K. Holden, and J. Butler. *Universal Principles of Design*. Rockport Publishers, 2003.
3. Nielsen, J. *Usability Engineering*. Academic Press, 1993.

<sup>2</sup>Clayton Lewis and John Rieman deserve the credit for this apt metaphor.

4. Standish Group. *The CHAOS Report*. Standish Group International, 1994 ([www.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www.standishgroup.com/sample_research/chaos_1994_1.php)).

*Whitney Quesenbery is a user interface designer, design process consultant, and highly regarded speaker. Ms. Quesenbery is an expert in developing new concepts that achieve the goal of meeting business, user, and technology needs. She has extensive user interface design experience and has worked on award-winning multimedia products, user interfaces, Web sites, and software applications.*

*Ms. Quesenbery is the owner and principal consultant of Whitney Interactive Design, LLC, where she works on projects from online financial news retrieval to hospital management software, Web applications, and corporate information tools for companies such as TriZetto Group, Novartis, Deloitte Consulting, Open University, McGraw-Hill, Siemens, ADP, Hewlett-Packard, and Eli Lilly. She has worked with many companies to implement a user-centered design process and was one of the key developers of LUCID, the Logical User-Centered Interaction Design framework, which supports integrating usability into a development process. She graduated from Bryn Mawr College and the National Theatre Institute.*

*Ms. Quesenbery can be reached at Whitney Interactive Design, LLC, 78 Washington Avenue, High Bridge, NJ 08829, USA. Tel: +1 908 638 5467; Fax: +1 908 638 5497; E-mail: whitneyq@wqusabilty.com; Web site: www.wqusabilty.com.*