

## Creating Software Users Will Love

Meaghan Beirne, Monique Gatner, and Dr. Melanie Rodney

**A solid design can go a long way in the success of a product. Similarly, a solidly designed software solution that provides a true user-friendly experience can also lead to a very successful product. This article looks at ways software developers (and system designers as well) can gain a much better understanding of how to focus a design around the user of a system rather than prioritize the needs of that system over the user.**

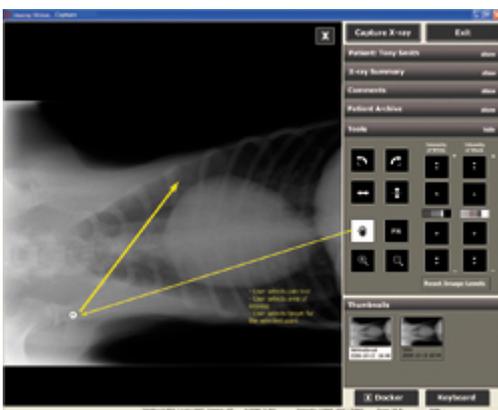
---

Meaghan Beirne, Monique Gatner, and Dr. Melanie Rodney work for Macadamian. Beirne is a usability specialist and can be reached at 613-820-2179 or [meaghan@macadamian.com](mailto:meaghan@macadamian.com) [1]. Gatner is a senior usability specialist and can be reached at 613-739-5976 x191 or [monique@macadamian.com](mailto:monique@macadamian.com) [2]. Dr. Rodney is a founding partner of the company and can be reached at 613-793-5976 x197 or [melanie.rodney@macadamian.com](mailto:melanie.rodney@macadamian.com) [3].

The runaway success of consumer devices like the iPhone have proven that great design sells. The iPhone has also raised customer expectations for all products including medical devices. Customers will no longer put up with poor design and subpar usability. They will punish companies that don't adopt good design practices, and reward those who do with repeat purchases and customer loyalty.

Great software is a large part of the iPhone's success. Today, many medical devices also have a large software component. Software can no longer be an afterthought, but what does it mean to have usable, inspiring software in a medical device? And how does one create great software when Apple's design team is not available?

### Usability in Software



How many times does someone start a task on their computer, only to forget how they did it last time and had to "guess" where to start? When users have experiences like these, they will guess incorrectly, waste a lot of time trying the wrong things, and ultimately, walk away or call support. These types of user

experiences are far too common in software.

While the term "user-friendliness" has traditionally been reserved for websites and desktop applications, users have become increasingly aware (and demanding) of user-friendliness from all devices—from their iPods to their car dashboards. Every industry of workers lays claim to highly unique and tailored devices that are made specifically for the needs of the employee; for example, cockpits are embedded with multiple devices made for pilots to be able to gain specific information, such as altitude and wind speed. In the same way, those in the medical profession are well-skilled at using devices made specifically for unique tasks, such as blood-oxygen monitors, heart wave monitors, and X-ray imaging machines. However, the radiologist who is highly skilled at using the X-ray imaging machine likely had to be trained on the device for a while, possibly under the help of a more senior staff member, just to be able to use it properly. This is not a reflection on the domain-specific skills of the radiologist, but a reflection of poor information design, buttons, and labels of the system, and a non-intuitive workflow.

How do so many devices end up with poor interface design? As a general rule, if you are close enough to a product to build it, you are too close to design it. Product developers have a tendency to design the "outside" (or the interface) of a product to reflect how it works on the "inside" (the internal structure of the program). However, most users do not need to know the algorithm of how their iPod selects the next random song; they simply want the songs to be mixed up. In the same way, the radiologist should not be forced to give an image a filename before they capture the image simply because the software needs a filename. Letting the user create the title after they have selected the best image that they will keep gives precedence to the user's task flow and not to the software's internal structure. Observing users in their environment, and testing early and often with prototypes, can mitigate risk and improve design to help ensure a successful product launch.

### **Talking to Users**

Giving precedence to users and their tasks will naturally make a product more usable. The only way to ensure this is to go and talk to the users—meet with them, talk with them, and watch them work.

By watching users interact with their devices, in their environment, it will offer a complete sense for how a product will fit in with their environment, or even how a product can help improve lagging or inefficient processes. One thing to remember is that all users will not have the same interaction needs for a piece of equipment. For example, nurses will typically interact with a respirometer in a minimal way—monitoring patient status, turning off alarms, and changing the display to check various measures for the therapy being administered. The respiratory technician, on the other hand, needs to interact with the system in a much more detailed way—setting parameters for the therapy to be administered, as well as accessing vital patient information. They also need to know how to quickly bypass the typical set up phase and go directly to ventilator start up in emergency situations. By observing different users in typical situations, designers can start to

## Creating Software Users Will Love

Published on Medical Design Technology (<http://www.mdtmag.com>)

---

develop an overall picture of system or device usage that allows them to tag critical tasks and make design decisions like:

- Which functions need to be immediately accessible?
- How is it ensured that critical functions are always performed without errors?
- How can monitored values be displayed for nurses to quickly assess patient status?

In non-traditional work environments such as hospitals and medical clinics, the user's environment can have a significant impact on how they interact with their devices. For example, a noisy work environment (or even noise from other devices) may obscure the auditory signal from a specific device; a visual signal or even a tactile signal, such as a vibration, may be needed to provide feedback to the user. Conversely, if the software helps the user by flagging issues of critical medical importance, the system must accomplish this in such a way to communicate the message to the user (operator) while not causing undue anxiety to the patient. The environment in which the device is used can be critical to whether it becomes an oft-used and heavily relied upon device, or whether it is quickly replaced by other, more reliable methods.

Context becomes highly important when working hospital-based applications. This was the case recently when creating a new software product that allowed nurses to monitor the location of pediatric patients with RFID bracelets. The nurses' environment and workflow were critical factors. In a high-interrupt and noisy environment, the monitoring application needed both auxiliary auditory and visual alerting. More importantly, significant focus was placed on understanding the task priorities; the most important tasks nurses would be performing in the application. The interface was streamlined and the monitoring screen was made to be the default view. The design was simplified further to ensure a high "signal-to-noise" ratio; in other words, so that the nurses could quickly identify the alarms, and the alarms showed only the critical information they would need in an emergency. All other tasks, such as administration of patient information, were placed on a separate tab, consistent with their lower priority. By paying extra attention to user tasks, the context in which the product would be used, and the priority of tasks, it was certain that the product would be quickly adopted by the target customer.

### Test Early, Test Often

Another component of designing for user's needs is to test the ideas early. It is a lot cheaper to make changes to a pencil and paper sketch, or a static graphic image, than it is to make changes to a functional system after hours of engineering time have been invested. Low-fidelity prototypes, such as paper and pencil drawings and images done on slides or captured in graphic design programs, can convey the basic information architecture and interaction design without having any of the components actually "working." Users can tell how they would attempt to complete

## Creating Software Users Will Love

Published on Medical Design Technology (<http://www.mdtmag.com>)

---

their tasks with the prototype, and designers are immediately able to determine whether they have structured the data and the task flow in a way that meets users' needs.

Users will also be able to communicate what is missing from the prototype; supporting users' tasks only achieves a partially usable system if there are major user needs that are missing. In recent testing conducted for a new touch screen design for a respirometer, respiratory technicians were walked through basic tasks for alternative designs. The designers gained a deeper understanding of how the technicians typically perform those tasks and how alternative interaction methods could support them in quickly accessing system features, like therapy modes and alarm settings, to complete their work more efficiently. This testing was conducted with static mock-ups of screens in various states illustrating different interaction methods such as buttons, menus, and tabs.

### Just Do It

Observing the end users and testing a design with them will help to ensure that a product will be adopted quickly and easily, and will help designers build a product that users can trust. The short amount of time spent up-front on the design will pay off later in reduced technical support, bug fixes, and lost revenues from poor user experience. Product development is like a poker game; you want to place small bets up front until you're sure you have a winning hand. The user-centered design process takes the risk out of design decisions to reveal the winning hand.

### Online

For additional information on the technologies and products discussed in this article, see *MDT* online at [www.mdtmag.com](http://www.mdtmag.com) [5] or Macadamian at [www.macadamian.com](http://www.macadamian.com) [6].

### Source URL (retrieved on 08/27/2014 - 1:07pm):

[http://www.mdtmag.com/articles/2009/03/creating-software-users-will-love?qt-video\\_of\\_the\\_day=0](http://www.mdtmag.com/articles/2009/03/creating-software-users-will-love?qt-video_of_the_day=0)

### Links:

[1] <mailto:meaghan@macadamian.com>

[2] <mailto:monique@macadamian.com>

[3] <mailto:melanie.rodney@macadamian.com>

[4] [http://www.mdtmag.com/sites/mdtmag.com/files/legacyimages/ProductImages/0903/capture-post2\\_large.jpg](http://www.mdtmag.com/sites/mdtmag.com/files/legacyimages/ProductImages/0903/capture-post2_large.jpg)

[5] <http://www.mdtmag.com>

[6] <http://www.macadamian.com>