Stimulated by regulations and standards\(^1\) as well as commercial imperatives, medical device manufacturers are striving to make medical devices safer by decreasing the potential for harmful use errors. Accordingly, manufacturers are observing and interviewing intended users about their interactions with devices en route to developing user interface requirements; applying human factors engineering (HFE) principles when designing user interfaces; and conducting formative and summative usability tests to improve and validate their devices' interactive quality. This has represented significant work for manufacturers, particularly the majority who started with little HFE knowledge and experience.

Today, the “awakening” to HFE in medical device development is essentially over—an established manufacturer who is just now discovering HFE has been out of touch. After all, the U.S. Food and Drug Administration (FDA) and AAMI conducted their first joint conference on HFE in 1995 when they previewed HFE expectations related to recent changes in quality system regulation (QSR).\(^4\) So, it has been 17 years since HFE entered the medical industry’s zeitgeist.

HFE specialists working in the medical industry used to focus on making devices more usable (i.e., user friendly). Since the QSR change, use safety has become the primary focus, resulting in user interfaces that are less vulnerable to harmful use errors.

It is rather obvious how HFE affects the safety of some medical devices. In the design of a surgical instrument, for example, HFE’s aim would be to reduce the chance of cutting anything other than the intended target. This goal could be accomplished by including a blade guard and a mechanical interlock that requires more than one deliberate action to enable cutting.

Guarding Against Use Error
The means to ensure software-user interfaces’ use-safety is arguably less obvious than the blade-guard example. However, medical devices with a software-user interface, which serve critical purposes and feature sophisticated functionalities, pose a design challenge that must be met. Although less self-evident than keeping a metal blade covered until the moment of intended use, guarding a software-user interface against harmful use errors can be relatively straightforward.

There are many ways to increase the use-safety of software-user interfaces. Some are explicit safety features, such as software algorithms that monitor users’ actions and try to prevent potential mistakes. Others are more implicit features, such as design characteristics...
User Needs

(dialogue styles, symbols, fonts, colors, information layouts, and demarcation lines) which facilitate clear and effective interactions with the device.

Together with a properly applied HFE process, such features combine to produce naturally safe software-user interfaces. Perhaps the most straightforward guard against a use error committed while interacting with a software-user interface is the ubiquitous confirmation dialogue:

Question: Are you sure you want to activate the pump?
Possible responses: “Yes” or “Cancel”

Device users have a love-hate relationship with these safeguards. Such safeguards can hinder a task, which actually is their purpose: getting users to think about the active task and the appropriateness of the step taken but not culminated.

However, safeguards can also lead to thoughtless behavior that undermines the safeguard’s purpose when users stop processing the question, “Are you sure you want to activate the pump?” and select “Yes” through the magic of muscle memory. Software-user interface safeguards can become less effective over time as users become inured to them, perhaps the same way they habituate to audible alarms.

What can be done about the problem that users can be their own worst enemy in terms of preventing harmful use errors? We see promise in smarter software-user interfaces with responsive safety mechanisms that guard against specific use errors.

An example of one of the earliest and most influential safety systems with a higher IQ is CareFusion’s Guardrails medication safety software installed on the company’s infusion devices, including the Alaris IV infusion pump.

Software-user interface safeguards can become less effective over time as users become inured to them, perhaps the same way they habituate to audible alarms.5

In essence, the software allows clinicians to program infusion devices to deliver medications at rates that fall within the healthcare institution-approved range (Figure 1).

If a nurse erroneously programs an infusion pump to deliver morphine at 10 times the hospital’s typical rate, the pump will prevent the infusion’s delivery. Hospitals can set custom ranges for individual drugs and tailor ranges to specific care settings, such as the neonate versus adult critical care unit in which safe drug delivery rates can differ.

Arguably, there is little difference between these safeguards and simple limits on a device’s operating range. What makes these safeguards “smart” is their consideration of differences in the patient population and the care settings.

Similar safeguards are now common in many medical devices. Many electronic health records (EHR) also include comparable safeguards, warning users about potentially hazardous situations by displaying “advisories”—messages indicating, for example, that a medication order conflicts with the institution’s standard practice (i.e., pharmacopeia). Healthcare providers seem to have also developed a love-hate relationship with such advisories, which can seem like unwelcome “speed bumps.”

Ultimately, safety features such as guardrails and EHR advisories can reduce the potential for use errors that software can detect. However, the potential remains for use errors and difficulties that result from fundamental design shortcomings, such as overly complex navigation, poor information layout, and lack of user guidance. This is why companies benefit from investing in the design of more “naturally safe” software-user interfaces, i.e., those with features that facilitate effective and, therefore, safe user interactions in the same manner as a scalpel blade guard.

Having observed thousands of usability test sessions, the authors can confidently state that legibility and readability of on-screen information are major safety factors. Factors such as character size, stroke (line thickness),

Figure 1. Screenshot of Sample Dose-Confirmation Software
foreground-background contrast, color, and spacing affect users’ ability to notice key information and read it correctly.

Figure 2 shows how a variation in fonts, weighting, and alignment can make a big difference in the legibility of patient vital signs. Such well-understood design principles are not always reflected in medical software-user interfaces. The root cause in many cases is the design team’s lack of interaction design expertise.

Given this root cause, the fundamental safety solution is to engage talented interaction designers in the software design process. Yes, the final risk control measure, a solution addressing poor usability test results, might be to use larger text on a given screen, for example. But the fundamental problem that would otherwise continue to plague the manufacturer’s products is the development team’s personnel. The naturally safe solution: legible and readable information as the by-product of assembling a design team that includes a talented interaction designer.

In the authors’ usability testing experience, guiding users through multi-step procedures decreases potential for safety-related use errors.

The HFE literature indicates that human beings have working memories that enable us to juggle 7 ± 2 items, and that our juggling ability can be affected by many performance shaping factors, such as stress and distractions.

The literature also tells us that people have imperfect recall of knowledge stored in long-term memory. What outcome would you expect when a therapeutic device requires a layperson to perform 19 steps in perfect order? If the

The Healthcare Technology Foundation, a 501(c)(3), was founded in 2002, on the principle that achieving improvement in the safe use of healthcare technology requires diverse stakeholders to come together in order to utilize their collective knowledge on the design, use, integration and servicing of healthcare technology, systems and devices.

The many issues surrounding human factors and information technology provide an excellent example of the need for such broad collaborations, and we are therefore enthusiastic in our support of this issue of Horizons.

Strategic initiatives for HTF can be found at
http://thehtf.org/
device relies on users to remember and execute all steps in perfect order, you could expect many use errors that might seem to result from forgetfulness or inattention. The real root cause is the burden that the lengthy procedure places on the user’s memory.

Conversely, if the device provides appropriate guidance at every step, one might observe almost perfect task performance in a usability test. The guidance might be a series of screens describing directed actions using well-written text and simple illustrations. The software-user interface might even present animations that match a given user’s preferred learning style—seeing a task performed instead of reading about it. In this case, the naturally safe solution is to guide users through a task rather than require them to recall a procedure. One might even provide a checklist at the end, noting that checklists have been demonstrated to be effective safety measures.

If a device does not provide necessary user support (i.e., guidance), the development team lacked an understanding of HFE (specifically the limits of human memory), lacked awareness of user guidance strategies, such as providing “wizard”-like support, or both. As in the first example of naturally safe software-user interface design, it seems the root cause of potentially harmful use errors is once again design team-related.

The authors have observed many test participants err when configuring a device for use, either by selecting menu options or entering data. Hypothetical examples include:

- Setting the device to “adult” mode when preparing to deliver therapy to a “pediatric” patient.
- Selecting the wrong dialysis therapy because the five-letter acronyms were almost identical.
- Selecting the wrong units (e.g., pounds instead of kilograms).

Such apparently simple problems are not unique to medical devices; a unit inconsistency error led the Mars Climate Orbiter to crash into the Red Planet instead of safely orbit it. These sample use errors might be avoided by applying established HFE principles. For example:

- The mode error might be avoided by designing the device to require users to select and then verify an operating mode instead of remaining in the previously selected mode or a default mode.
- The dialysis therapy selection error might be avoided by spelling out the modes in addition to, or in place of, the acronyms.
- The unit selection error could be mitigated by the software checking for unit consistency, displaying the units in larger-than-normal text, or asking the user to confirm the units of measure (Figure 3).

These are certainly appropriate solutions that would make a user interface more naturally safe, but it takes a team with HFE capabilities to recognize and realize such opportunities. Notably, capable interaction designers help resolve usability issues that are less straightforward than those described previously. Perhaps most importantly, interaction designers focus on the software-user interface’s conceptual model—the organization scheme behind the user interface’s features and controls. A coherent conceptual model (i.e., one that matches users’ expectations) makes the difference between a device that users consider inherently intuitive and one that users repeatedly struggle to use.

**Natural Safety Through Comprehensive HFE**

Near the end of the development process, you can assess how naturally safe a software-user interface is by conducting a summative usability test. If test participants perform tasks smoothly, the user interface is naturally safe. Conversely, if participants make many mistakes, including some that remain uncorrected, there is a problem. Chances are that such a product might undergo some “tweaking” in the form of last-minute screen changes that do not
require major software revisions. The developers can then focus on revising the product’s instructions for use (IFU) and training.

Last-minute and minor software “tweaks,” along with modifications to the IFU and training, suggest that a product is not naturally safe and point to shortcomings in the development process. As such, the root cause of many use errors, some of which could be harmful, is inadequate application of HFE, 17 years after the QSR required manufacturers to identify user needs and then that medical products meet them.

Applying HFE and controlling use-related risk are inexorably linked. Applying HFE to medical software-user interfaces is the ultimate risk control measure that leads to specific safety enhancements, such as properly-sized text, clear procedural guidance, and deliberate mode selection.

References
2. ANSI/AAMI 62366:2007, Medical devices—Application of usability engineering to medical devices. Association for the Advancement of Medical Instrumentation, Arlington, VA.

A coherent conceptual model (i.e., one that matches users’ expectations) makes the difference between a device that users consider inherently intuitive and one that users repeatedly struggle to use.
Human Factors Resources

**Handbook of Human Factors in Medical Device Design**
Written as a compliment to HE75, this handbook includes expanded discussions of design issues and product design case studies. Each chapter stands alone and includes tables, illustrations, and cross references, allowing you to quickly find the exact information you need. This handbook was not developed by the AAMI committee.
844 pages
Order Code: HFMDD
List $170 / AAMI member $115

**Human Factors and Ergonomics in Health Care and Patient Safety, 2ed**
The 2nd edition of this book is the ultimate compendium of human factors and ergonomics issues relevant to healthcare and patient safety. It provides a practical approach with coverage of methods, interventions, and applications. Topics include work schedules, error recovery, telemedicine, workflow analysis, simulation, health IT development and design, and patient safety management. 876 pages.
Order Code: HFEH
List $180 / AAMI member $125

**Patient Safety: A Human Factors Approach**
This in-depth practical publication covers a wide array of issues facing the human factors field and its importance for patient safety. Designed to provide advice for all levels of professionals in healthcare, the book includes the latest research with examples that connect theory to actual practice. The author, Sidney Dekker, presents information in easy-to-use bulleted lists and illustrations where possible. 261 pages.
Order Code: HFA
List $45 / AAMI member $32

Order Your Copy Today!
Call +1-877-249-8226
Visit http://my.aami.org/store

SOURCE CODE: PB