

Chapter 5 / Capítulo 5

Primum non nocere: Software engineering
for medicine
ISBN: 978-9915-9876-8-2
DOI: 10.62486/978-9915-9876-8-2.Ch05
Pages: 79-94



© 2026 The authors. This is an open access
book chapter distributed under the terms of
the Creative Commons Attribution (CC BY)
4.0 License.

Chapter 5. IEC 62366: Human Factors and Usability

Annier Jesús Fajardo Quesada  

Eduardo Antonio Hernández González 

René Herrero Pacheco 

Rolando Javier Álvarez Pérez 

1 Granma University of Medical Sciences. Granma, Cuba.

2 University of Medical Sciences of Pinar del Río. Pinar del Río, Cuba.

“No matter how perfect an algorithm may be, no matter how rigorously it has been developed, if the physician using it cannot understand it, if the patient does not know how to interpret it, if in an emergency the interface becomes an obstacle, then all the effort will have been in vain. Technology does not exist in a vacuum: it exists to be used by people, and it is those people who, ultimately, determine its success or failure.”

Imagine an operating room in the middle of a complex procedure. The surgeon needs quick access to the patient’s information, but the monitor screen is cluttered with irrelevant data. He tries to adjust the settings, but the menus are confusing, and the buttons are too small for his gloves. He asks the nurse for help, but she cannot find the right option either. Seconds tick by—seconds that are precious in surgery. Finally, someone manages to access the information, but the delay has created tension and, worse still, has broken the team’s concentration. This unfortunately common scenario illustrates an uncomfortable truth: technically perfect software can be clinically useless—or even dangerous—if it does not consider the people who will be using it. That is exactly what usability engineering and the standard that governs it, IEC 62366, are all about.

IEC 62366 is the international standard that establishes the requirements for the usability engineering process applied to medical devices. Its full title is “Medical devices—application of usability engineering to medical devices,” and its objective is to ensure that products are safe and effective when used by the people for whom they are intended in the environments where they are actually used.⁽¹⁾ Unlike other standards that focus on the internal workings of software, 62366 looks outward, toward the interaction between the product and its users. In addition, it does so with a deep conviction that usage errors are not the user’s fault but rather the fault of the design and that preventing those errors is the manufacturer’s responsibility.

In this chapter, we explore the fundamentals of usability engineering as applied to medical software. We will see how to integrate it from the earliest stages of design, how to identify usage-related risks, how to evaluate the interface with real users, and how to document the entire process. However, we start with the most important thing: understanding that usability is not an aesthetic embellishment or a superficial layer added at the end but a fundamental dimension of safety. Software that is difficult to use is not just a nuisance; it is a risk to the patient.

5.1 USABILITY ENGINEERING FROM THE FIRST SKETCH

“Usability is not a superficial layer, but a deep-seated property that emerges from design decisions made throughout the entire process.”

There is a widespread misconception in the world of software development, especially among those coming from unregulated environments, who view usability as something added at the end of the project, when the functionality is already implemented and all that remains is to “polish” the presentation. A graphical interface designer is hired and asked to “beautify” the application with attractive colors and modern fonts, and it is expected that with a few cosmetic tweaks, the product will become intuitive and userfriendly. This approach, which is already questionable in any type of commercial software, is simply dangerous and potentially lethal in the medical field. Because usability is not a superficial coat of paint but a deep, structural property that emerges from the design decisions made throughout the entire development process. If it is not considered from the very beginning—when requirements are being defined and the system architecture is being outlined—the final result is very likely to be a product that, although it functions correctly from a technical perspective, is difficult or impossible to use safely in actual clinical practice.

Usability engineering, as rigorously defined by IEC 62366, is a systematic and planned process that must begin in the earliest stages of development, long before the first line of code is written. It begins with a thorough identification of the product’s users: not only medical specialists but also nurses on different shifts, maintenance technicians, patients who may interact with the system, and family members who sometimes act as informal caregivers. Each group has different demographic characteristics, distinct clinical needs, widely varying levels of technological training, and specific usage contexts that must be deeply understood. Software intended for use by medical specialists in a controlled hospital setting can afford a certain level of complexity and information density; software intended for use by elderly patients in their private homes must be extremely simple, intuitive, and error-tolerant. Getting to know the actual users, understanding their physical and cognitive limitations as well as their real capabilities, is the essential first step toward designing something that truly serves them in their living or working context.

5.1.1 CONTEXT OF USE, PROTOTYPES, AND EARLY ITERATION

The next methodological step is to gain a deep understanding of the actual context of use, which rarely aligns with designers' idealizations. Using software in a quiet primary care office, with ample time and no interruptions, is not the same as using it in a bustling intensive care unit, with alarms constantly ringing, distressed family members asking questions, and life-or-death decisions to be made in seconds. It is not the same to use it in a well-lit hospital with stable infrastructure, as it is in a private home with dim lighting, reflective screens, and an intermittent internet connection. It is not the same to use it hands-free and sitting comfortably, as it is while it is wearing latex gloves during a sterile procedure or in the middle of a 24-hour night shift when physical and mental fatigue clouds one's attention and decision-making.

Usability engineering requires designers to literally immerse themselves in these contexts, to observe without interruption, to ask questions with humility, and to understand the real and often adverse conditions in which the product will be used. Only then can they anticipate usability issues and design solutions that work in real clinical practice, not just in the development laboratory under ideal conditions.

With these collected ethnographic data, the actual interface design phase begins. However, the goal is not to design the polished final version directly but rather to quickly create prototypes, sketches, and mockups that can be evaluated with real users before investing scarce time and resources in full-scale development. A prototype can be as simple as drawings on paper that simulate screens or as elaborate as an interactive simulation that responds to user actions. What matters is not the degree of technical sophistication but that it allows ideas to be tested with real users quickly and cost-effectively before they commit to decisions that are difficult to reverse.

This iterative approach—involving rapid, low-cost trial and error—is a hallmark of usability engineering and stands in stark contrast to the traditional approach of designing everything in detail first and testing only at the end, when changes become prohibitively expensive. Testing early on, testing frequently with different users, and making adjustments on the fly on the basis of real evidence constitute the most efficient and reliable way to achieve a truly usable interface in the clinical setting.

The participation of real users in these prototype tests is absolutely irreplaceable, no matter how uncomfortable it may be for designers. Regardless of how hard design professionals try to put themselves in the shoes of a medical professional or patient, cultural, practical, or cognitive aspects escape them because of differences in training and context. Only by directly observing a real clinician interacting with a prototype do we discover that the button that seemed obvious and wellplaced was not so obvious to someone in a hurry that the screen flow that seemed logical from a technical perspective did not match the actual sequence of clinical work and that the medical terminology used by engineers was not exactly what clinicians routinely use in their daily practice.

These observations, although they sometimes humble our design assumptions, are pure gold for improving the product and must be systematically documented and used to iterate on the next prototype. Usability is not designed in an ivory tower of experts; it is built through continuous and respectful dialog with those who use the product in their real lives.

A crucial aspect of this process that distinguishes it from conventional interface design is the specific identification of risks related to the product's use. IEC 62366 formally requires a systematic analysis of potential usage errors and their possible consequences for the patient. For example, if a critical action button is too small on a touchscreen, there is a predictable risk that the user will accidentally press the wrong button in a stressful situation. If a clinical alert is not sufficiently visually and audibly prominent, there is a risk that it will go unnoticed in a noisy environment. If the sequence of steps to complete a task is confusing or unintuitive, there is a risk that the user will complete the steps in the wrong order or omit a critical one.

Each of these usability risks must be explicitly identified, assessed in terms of potential severity, and, if necessary, mitigated through improvements to the interface design. Usability in the medical context is not just a matter of convenience or user satisfaction but of patient safety.

5.1.2 HIERARCHIES OF CONTROL, USER DIVERSITY, AND DESIGN CULTURE

The hierarchy of control measures we saw in technical risk management also applies here to usage risks, with the same priority logic. The best way to prevent a usage error is to design the product so that the error is physically or logically impossible to make.

For example, if we do not want the user to enter an incorrect medication dose, we can limit the values allowed by the software to a clinically established safe range so that even if the user attempts to enter an out-of-range value by mistake, the system simply does not accept it and prompts for reconsideration. If we cannot make the error impossible for reasons of clinical flexibility, we can at least make it difficult to commit inadvertently, for example, by requiring conscious double confirmation for critical actions that cannot be easily undone. If we cannot make it difficult, we can at least make it visible and recoverable, for example, by clearly showing the consequences of the action before it is definitively confirmed and allowing it to be undone within a reasonable time window. Safety information, such as generic on-screen warnings or notes in the manual, is the last line of defense and should be used only when the previous design measures are insufficient because of technical or clinical limitations.

Documentation of the entire usability engineering process is another formal requirement of the standard that serves an important practical purpose. There must be a written record of exactly who the intended users are and their characteristics, what the detailed context of use is, which prototypes were developed in each iteration, which usability tests were conducted with whom and under what conditions, what specific usability issues were identified, and how they were resolved through design changes. This documentation is not meaningless bureaucratic red tape but tangible evidence that the process was carried out correctly and with due diligence. In the event of an incident related to the use of the product, it will be the first thing regulators and potentially the courts examine to determine whether the manufacturer acted responsibly.

If the documentation demonstrates a rigorous process, with early identification of usability issues and control measures implemented progressively, the manufacturer will be in a much stronger position than if everything was done haphazardly or without evidence.

For developers accustomed to working in other, less regulated fields, systematic usability engineering can represent a significant cultural shift that requires adaptation. It involves accepting that one's own design work will be observed, constructively criticized, and modified on the basis of the opinions of people who are not experts in technology or design. It involves setting aside pride in one's own solution when user testing shows that it does not work as expected. It involves understanding that the ultimate goal is not to demonstrate how clever or elegant one is as a designer but to build something that effectively serves others in the context of their needs. This intellectual humility is difficult for technical professionals proud of their abilities, but it is essential for those who wish to seriously dedicate themselves to medical software. Unlike in other fields of software development, an interface design error does not simply mean a bad review in the app store or a complaint on social media but potentially a harmed patient or a life lost.

For doctors and other healthcare professionals involved in these design processes, usability engineering is an invaluable opportunity to contribute their most valuable and contextualized knowledge: that of real-world, chaotic, and imperfect practice. They can pinpoint exactly why a certain screen flow does not match how they actually work in their department, why technical terminology is not what they typically use in their daily communication, or why a feature that seems useful in theory is rarely used in practice because of time constraints or prioritization. They can actively participate in prototype testing, candidly pointing out what they find confusing, what they consider unnecessary, and what they feel is missing for their work.

One aspect that is often underestimated, even by well-intentioned teams, is the actual diversity of users within a single clinical setting, which necessitates designing for the "reasonable worst-case scenario." A physician with twenty years of experience in the department is not the same as a resident who has just arrived and is still receiving the hang of the protocols. A day shift nurse, with the full support team available, is not the same as a night shift nurse, alone on the floor, with limited resources and accumulated fatigue. A young, tech-savvy patient is not the same as an older patient with vision problems, reduced manual dexterity, or a fear of technology. Usability engineering must explicitly consider this diversity and design for all of them, not just for the ideal, expert user under optimal conditions. A product that only works for the expert user under ideal conditions is a product that will fail in real-world practice because clinical practice is full of diverse users in diverse and often adverse conditions.

The rapid evolution of technology also poses fascinating yet complex new challenges for medical usability. Multitouch interfaces, voice commands, augmented reality, and wearable devices open up new possibilities for interaction but also introduce specific new risks that must be managed. A voice command misinterpreted because of ambient noise can have serious consequences if it activates a critical function without confirmation. A touch interface with buttons that are too small or too close together can lead to touch errors in stressful situations. An augmented reality system that overlays visual information in the surgeon's field of view can

distract them rather than help them if it is not carefully designed. Usability engineering must keep pace with these emerging technologies, assessing their specific risks of use and developing innovative methods to mitigate them without sacrificing their benefits. It is not a matter of conservatively rejecting the new out of fear of change but of incorporating it with scientific prudence and rigorously testing it before relying on it for patient care.

It is worth remembering that usability in the context of medical software is not an aesthetic luxury or a commercial gimmick but a fundamental aspect of patient safety. Software that is difficult to use, confusing, or frustrating is not just an annoyance for the professional; it is an active risk to the patient. When the user becomes frustrated and when they cannot quickly find what they are looking for, when they hesitate to interpret critical information or how to perform an urgent action, the probability of error increases exponentially. In medicine, errors have real and often irreversible human consequences. Usability engineering, rigorously applied from the very first sketch and maintained throughout the entire lifecycle, is the best tool available to prevent such usage errors. It does not absolutely guarantee that they will not occur, but it makes them far less likely. In a field where the probability of error directly translates into human lives affected, any reduction—no matter how small it may seem statistically—is worth every effort invested.

5.2 USAGE ERRORS VS. USER ERRORS

Imagine we are in an intensive care unit, in the middle of a hectic and noisy night. An experienced nurse, who has been on duty for 12 hours and is visibly fatigued, urgently needs to program an infusion pump to administer a critical vasoactive drug to stabilize an unstable patient. The pump's interface displays the options on a touchscreen, but the buttons are small and closely spaced, and the tactile feedback is imprecise. With fingers trembling from physical exhaustion and emotional pressure, the nurse accidentally pressed the button next to the one she actually intended, unwittingly selecting a dose ten times higher than what the doctor prescribed. The pump begins to infuse the medication at that dangerous rate, and it is not until several minutes later, when the patient begins to show alarming signs of severe hypotension, that someone realizes the error and reverses the situation. Whose fault is this incident that nearly cost the patient their life? The nurse's, for not being careful enough at a critical moment? The manufacturer, for designing an interface that facilitates and almost invites errors under stressful conditions? The hospital, for having such long shifts that inevitably lead to chronic staff fatigue? The answer to this question is not merely academic or philosophical: it determines how the incident will be formally investigated, what corrective measures will be implemented to prevent recurrences, and above all, how future patients will be protected from similar situations.

The distinction between usage error and user error is among the most important and simultaneously most misunderstood concepts in all usability engineering applied to medical contexts. A user error is fundamentally attributed to a lack of attention, specific knowledge, or technical skill on the part of the individual using the product at that moment. It is viewed as a personal failure, a deviation from the expected ideal behavior. A usage error, on the other hand, is caused, facilitated, or amplified by characteristics of the product's design itself, by interface decisions that did not adequately consider human limitations.

The difference is subtle on the surface but crucial in its practical implications because it determines where we focus our intervention to prevent future incidents. If we consider what happened to be an isolated user error, the solution will be to better train staff through additional courses or to discipline the nurse as a deterrent or simply to verbally remind her to be more careful in the future. If we consider it a systematic usage error, the solution will be to redesign the interface so that the error is physically more difficult or even impossible to make, eliminating the condition that causes it. The first approach barely reduces the likelihood of it occurring again to another user in another stressful situation; the second eliminates the possibility of that specific error at its root for all future users.

5.2.1 DESIGN RESPONSIBILITY AND TASK-CAPABILITY MISMATCH

IEC 62366 takes a clear and explicit stance regarding this distinction: usage errors are the responsibility of the medical device manufacturer, not the user. The standard formally requires that the usability engineering process proactively identify potential errors users might make given the proposed design, assess the potential consequences of those errors for the patient, and

take design measures to prevent them or at least mitigate their effects. This does not mean, of course, that users are considered infallible or that manufacturers must absolutely guarantee that no errors will ever be made. This means that the design must be inherently tolerant of human error, recognizing from the outset that the people who will use the product become physically and mentally fatigued, are distracted by multiple stimuli, make mistakes under pressure, and that the product must be explicitly prepared for these inevitable situations.

Good medical design does not assume that the user will be perfect, attentive, and expert; it assumes that the user will be human with all their limitations and designs accordingly to protect the patient from those limitations.

To systematically distinguish between usage errors and user errors in incident analysis, human factor experts use the technical concept of a “mismatch” between the cognitive and physical demands of the task and the actual capabilities of the typical user under real-world working conditions. When a task consistently demands more than a typical user can provide under normal conditions of use in the clinical environment, the resulting error is by definition a usage error, not an error of the individual. For example, if a software interface requires a complex sequence of steps that are only used in rare emergency situations, it is predictable that in a real crisis situation, the user, no matter how competent, will forget some critical step because of stress. This is not a user error due to carelessness; it is a design error for failing to provide mnemonic support or simplify the workflow.

If an alarm is so frequent that users end up ignoring it out of habit and therefore fail to respond to a truly critical alarm, that is not user negligence for failing to pay attention; it is a design flaw for failing to properly calibrate the system’s sensitivity. The burden of responsibility should not fall on the user to compensate for the product’s shortcomings; it is the product that must adapt to human capabilities.

Fatigue is another critical factor that turns seemingly individual errors into predictable systemic design failures. A product that functions acceptably with rested users under laboratory conditions can become dangerously unsafe with fatigued users on night shifts. Given that chronic fatigue is endemic in modern healthcare settings—with shifts of twelve hours or more, nights without adequate sleep, and high emotional and physical workloads—designers must assume as a design condition that their users will frequently be fatigued and explicitly design accordingly.

This means simpler interfaces with fewer simultaneous options, mandatory confirmations for critical irreversible actions, and interface elements large enough and high-contrast enough so that they do not require exquisite visual attention and workflows that minimize the cognitive load during times of stress. A design that works only with perfectly rested and attentive users is a design that will inevitably fail in real clinical practice, where fatigue is the norm, not the exception.

The user’s expected prior knowledge is another determining factor in this classification. A physician who is an expert in their clinical specialty may have no significant knowledge of computing or touch interfaces. An elderly patient may have no medical or technological knowledge whatsoever. Assuming that the user possesses certain technical knowledge that they do not actually have is a predictable recipe for user error. That is why the design must be rigorously based on the actual, documented profile of the intended users, not on an idealized profile imagined by engineers. It must use language they can understand without a dictionary, follow workflows that feel natural given their work habits, and provide contextual help when necessary without being condescending.

User training can complement good design, but it can never completely replace an intuitive and self-explanatory design. If a medical product requires a training course lasting several hours to use it without making serious errors, the fundamental problem is likely not a lack of user training but rather the unnecessary complexity of the design that does not respect human cognitive limitations.

5.2.2 LEGAL IMPLICATIONS, LEARNING FROM ERRORS, AND FORESEEABLE USE

The distinction between an error of use and a user error has significant legal and regulatory implications that manufacturers must understand. In many developed legal systems, if an incident is considered an isolated user error, legal liability falls primarily on the healthcare professional or the employing institution. If it is considered a systematic error of product use—liability may fall on the manufacturer for a design defect. That is why some manufacturers, in a short-sighted and ethically questionable manner, resist acknowledging that recurring errors with their products are design flaws, preferring instead attributing them to individual user inexperience or exceptional

circumstances. However, this defensive attitude, in addition to being ethically reprehensible because it puts patients at risk, is commercially short-sighted. If the design is not fundamentally improved, errors will continue to occur with different users in different places, and sooner or later, some independent researchers or regulators will statistically demonstrate that the error pattern is systematic and that the root cause lies in the product, not in the people who use it. At that point, the manufacturer's legal and reputational liability will multiply.

For development teams, the distinction between usage errors and user errors must be actively present in every phase of the design, becoming a standard lens for analysis. In formal testing with representative users, when someone makes an unexpected error, the question the team should ask is not "Why is this user so clumsy or distracted?" but "What is it about our current design that led to this predictable error?" This simple shift in perspective—from judging the user to analyzing the system—transforms usability testing from a superficial validation exercise into a genuine source of deep learning.

Every error identified during testing is a valuable opportunity to improve the design before it causes real harm, not a personal failing to be hidden or downplayed. Teams that consistently adopt this mindset of systemic analysis develop products that are inherently safer because they have been shaped and refined by the real errors of real users under controlled conditions, not by the theories of isolated designers.

A related and equally important concept is that of "reasonably foreseeable abnormal use," which the standard explicitly incorporates. The standard recognizes that real users sometimes use medical products in ways not explicitly anticipated by the manufacturer in its manuals. They may skip mandatory steps when in a hurry, use the product for purposes slightly different from those authorized, combine it with other devices in unauthorized but practical ways, or find undocumented shortcuts that save time. These behaviors, while not ideally anticipated, are statistically foreseeable given human nature and the pressures of the clinical environment, and the manufacturer must actively consider them in its analysis of risks of use. If it is foreseeable that a user, frequently in a hurry, will attempt to speed up a process by skipping safety confirmations, the design must technically prevent that from being possible, not merely warn against it. If it is foreseeable that a patient at home will manipulate the device in unauthorized ways out of ignorance or desperation, the design must be robust against such foreseeable manipulation. Foreseeable abnormal use is not a legal excuse for the manufacturer; it is part of the context of use that must be considered and managed as a risk.

Documentation of usage errors identified during development and the measures taken to prevent them is a mandatory part of the usability dossier required by IEC 62366. It must include a comprehensive list of potential errors identified through systematic analysis, their estimated severity for the patient, their probability of occurrence under real-world conditions, and the design control measures implemented for each. It must also include the results of empirical user testing, showing which errors actually occurred during the test sessions and how they were resolved through design iterations. This documentation is valuable not only for the regulators who will audit the product but also for the development team itself because it constitutes an organizational record of the decisions made and the lessons learned in the process. In long-term projects spanning years, with inevitable staff changes, this documented record prevents the same design errors from being repeated repeatedly with each new generation of developers.

For physicians and other healthcare professionals collaborating on these design projects, understanding the distinction between usage errors and user errors helps them participate more effectively and constructively in usability testing and the early identification of problems. Drawing on their accumulated experience, they can identify which errors are likely to occur in real-world practice even if they do not occur during controlled laboratory testing. From their clinical perspective, they can help distinguish between what is an occasional error attributable to a specific user and specific circumstances and what is a systematic pattern attributable to design flaws that will recur. Their perspective is essential for usability analysis to be realistic and not limited to what happens in the laboratory under artificial conditions. A physician who has spent years on night shifts can anticipate usage errors that a young designer would never imagine.

One aspect that often generates technical and ethical controversy is the dynamic balance between functional flexibility and patient safety. A highly restrictive design, which prevents any deviation from the intended workflow, can be very safe but very frustrating for expert users who need efficiency and adaptability in unforeseen situations. By allowing too many options, a highly flexible design can be efficient and adaptable but also prone to usage errors. There is no single universal answer; it depends on the specific clinical context, the intended user type, and the criticality of the tasks being performed. What the standard requires is that this tension be explicitly

addressed in the design process, that decisions be informed by evidence and that the reasons for those decisions be documented for future reference. Usability is not about maximizing a single variable but about finding the right balance for each specific usage situation.

Usage errors are not limited to those made by the clinical end-user or patient. They can also be made—with equally serious consequences—by those who install the software, maintain the hardware, or even clean the product in the hospital setting. An installation technician who misconfigures software because the instructions are confusing or incomplete; a maintenance professional who inadvertently disables a safety alarm because they do not know how to temporarily silence it correctly; and a cleaning assistant who wets a sensitive device because it is not clearly marked as vulnerable to water. All of them are users in the broadest sense of the term, and their errors must also be systematically considered in usability analysis and risk management. Patient safety is a chain of responsibilities that extends throughout the entire product lifecycle, and it is only as strong as its weakest link, which can be at any point in that chain.

5.3 EVALUATION WHILE WEARING GLOVES, SLEEPDEPRIVED, AND WITH INTERRUPTIONS

Usability evaluation in controlled laboratory settings produces clean, orderly, and deceptively optimistic results. When a designer observes a physician interacting with a prototype in a quiet room, free from auditory distractions, with ample time and the user well-rested and attentive, they inevitably obtain an idealized picture of human behavior—which is rarely replicated in reality. That picture is useful for identifying obvious, serious design problems, but it does not reveal how the product will behave under the real-world conditions in which it will be used—which are exactly the opposite: chaotic, noisy, interrupted, and exhausting. Real-world clinical practice is characterized by constant interruptions that fragment attention, accumulated fatigue that clouds judgment, physical limitations such as latex gloves or poor nighttime lighting, and constant time pressure that leads to cognitive shortcuts. Evaluating under these adverse conditions is not just one methodological option among many; it is an absolute necessity for patient safety.

IEC 62366 explicitly requires that usability evaluation be conducted under conditions that adequately represent the intended real-world usage environment, not just under ideal laboratory conditions. This means that formal laboratory testing with rested, attentive, and motivated users is not sufficient. Test scenarios must be deliberately designed to incorporate the stressors typical of the clinical environment: constant background noise, scheduled interruptions, forced multitasking, induced or natural fatigue, and real physical limitations. Only in this way can we discover how the product will actually behave when it is in the hands of those who truly need it in their daily work. A product that works perfectly in the laboratory but consistently fails in clinical practice is a product that has been poorly evaluated by design, no matter how rigorous the formal testing may have appeared to be.

5.3.1 PHYSICAL LIMITATIONS AND FATIGUE: THE BODY IN THE CLINICAL SETTING

Evaluation while wearing gloves is one of the most concrete, revealing, and frequently overlooked aspects of real-world clinical usability. In many hospital settings, professionals are required to wear latex or nitrile gloves, which significantly reduce tactile sensitivity and hinder the precise manipulation of small interfaces. A touch interface that works perfectly with a bare, dry finger can become unmanageable or dangerous with wet gloves or standard-thickness gloves. Buttons that are too small on the screen, sliders that require pinpoint precision, and complex multitouch gestures can become frequent sources of error when the user does not feel the contact well or when the glove rubs against the surface. Systematic evaluation with gloves is not a methodological luxury; it is a necessary condition for any product intended for use in environments where protecting staff from infection is mandatory and nonnegotiable.

Fatigue is another critical factor that is rarely genuinely incorporated into conventional usability evaluations, even though its impact is devastating. Healthcare professionals work long shifts of twelve hours or more, often at night, with accumulated hours of wakefulness that objectively affect their cognitive and motor abilities. A fatigued person has slower reaction times, makes more errors of omission because of carelessness, tends to rely on cognitive automatisms rather than carefully processing new information, and has reduced divided attention capacity. A product that requires constant sustained attention, presents information in a complex or dense manner, or demands critical decisions at moments of peak fatigue will be particularly prone to

usage errors when used under real-world night shift conditions. Testing with genuinely fatigued users, even if logistically difficult to organize, provides vital information that cannot be obtained through any other form of evaluation. Some teams even induce controlled fatigue during test sessions to simulate these effects.

5.3.2 INTERRUPTIONS, MULTITASKING, AND TIME PRESSURES

Interruptions are the absolute norm in contemporary clinical practice, not the occasional exception that designers would prefer to ignore. Doctors may be interrupted dozens of times during a consultation or procedure by calls, alarms, or questions from colleagues or family members. A nurse may have to handle multiple tasks simultaneously in different rooms, switching from one to another without being able to complete any of them from start to finish because of the urgency of the demands. A software product that is not explicitly designed to handle interruptions—one that loses its task state when the user is briefly away or that forces users to restart processes from the beginning without saving partial progress—becomes a source of intense frustration and systematic error. Usability evaluation must include scenarios with scheduled interruptions, where the user has to resume incomplete tasks after a distraction, remember exactly where they left off, and retrieve information they may have lost from working memory. Only then can the true robustness of the design be evaluated in the face of fragmented clinical reality.

Multitasking is a natural and constant extension of interruptions in healthcare work. Professionals rarely perform a single task from start to finish without deviations; rather, they simultaneously manage a set of tasks that overlap in time, switching between them according to the changing priorities of the moment. A product that assumes an ideal linear workflow, with sequential steps that must be completed strictly in order without deviations, clashes violently with this cognitive and operational reality. The evaluation must include scenarios where the user must explicitly manage multiple concurrent tasks, switch between them frequently, and maintain mental coherence for each without becoming confused. This reveals whether the design supports the inherent cognitive complexity of real-world practice or, conversely, adds an additional burden that increases the risk of error due to attentional overload.

Compared with relaxed laboratory conditions, real-world time pressure is another factor that profoundly distorts human behavior. When time is short because of clinical urgency, users skip verification steps, ignore information they consider secondary, and make quick decisions on the basis of recognition heuristics rather than systematic analysis. A product that appears safe and well designed under calm laboratory conditions can become dangerously unsafe when used under real-world time pressure. Safety confirmations that, in theory, should prevent critical errors may be accepted mechanically without actually reading the content, alarms may be silenced quickly without processing their meaning, and double-checking procedures may be skipped because of overconfidence in the technology.

The evaluation must include scenarios with realistic and demanding time limits, forcing users to prioritize quickly and take cognitive shortcuts. This reveals which parts of the design are robust under time pressure and which become sources of risk when the user cannot devote the attention the designer assumes.

5.3.3 PHYSICAL ENVIRONMENT, HUMAN DIVERSITY, AND INVESTMENT IN SAFETY

Poor lighting in patient rooms at night to avoid disturbing rest, intense ambient noise from multiple devices that masks important alarms, limited physical space around the bed that hinders equipment handling, and extreme temperatures in certain environments such as operating rooms or burn units. A product that has not been systematically evaluated under these adverse conditions may fail precisely when its proper function is most needed. The evaluation must reproduce, as far as possible, these real-world environmental conditions or at least explicitly consider how each affects the specific use of the product. An audible alarm that is clearly audible in a quiet laboratory may be completely inaudible in a noisy ICU; a display that is legible under controlled ambient lighting may be illegible in direct sunlight or under glare from surgical lamps.

The actual physical diversity of users is another factor often overlooked in standard evaluations, limiting the generalizability of the results. Not all users have the same visual acuity, the same fine motor skills, the same hearing ability, or the same grip strength. Healthcare professionals, like any human population, include people with visual impairments who wear glasses or contact lenses,

people with essential tremors or mild Parkinson's disease, and people with hearing problems caused by occupational factors or aging. A product that assumes the "standard" sensory and motor abilities of a young, healthy user effectively excludes a significant portion of its potential users and, worse still from a safety perspective, can lead to systematic errors when those users with limitations are forced to use it without adaptations. Evaluation must deliberately include users with diverse functional abilities or at least explicitly consider, through task analysis, how common limitations in the population affect the safe use of the product.

Cultural and linguistic diversity also matters significantly in globalized and multicultural healthcare settings. Iconographic symbols that are intuitive in one culture may be confusing or even offensive in another. Colors that have clear and consistent meanings in one cultural context may have different or contradictory meanings in another. Interface metaphors that work perfectly in one language may not translate well or may create semantically false friends in another. A product intended for use in multicultural settings—as is often the case in modern healthcare—must be evaluated with users from diverse cultural and linguistic backgrounds. This does not necessarily mean designing radically different versions for each culture, but it does mean ensuring that fundamental navigation and safety elements are universally understandable and that there are no cultural biases in the design that could lead to critical misinterpretation.

Specific evaluation with patients, not just with healthcare professionals, adds another layer of complexity and necessity. Patients who use medical devices at home have very different characteristics from trained professionals do. They may have cognitive limitations associated with their underlying condition, age-related vision or hearing problems, or motor difficulties due to neurological conditions. They may be anxious about their diagnosis, frightened by the procedure, or suffer from pain that distracts their attention. They may have no technical or medical training whatsoever and rely blindly on what the device tells them. A product that is easy for a trained professional to use may be completely incomprehensible or dangerous for a vulnerable patient.

For development teams, organizing evaluations under these realistic conditions is undoubtedly complex and costly and consumes valuable resources. It requires recruiting users with the appropriate characteristics who are willing to participate under adverse conditions, replicating physical environments that simulate clinical reality with a certain degree of fidelity, and designing test scenarios that convincingly incorporate relevant stressors. However, the alternative is worse in every way: launching a product onto the market that has not been tested under real-world conditions and discovering, through clinical incidents resulting in patient harm, that it fails precisely when its reliability is most needed. The economic and reputational cost of a comprehensive realistic evaluation is always lower than the human and legal cost of a subsequent clinical failure. Investing in realistic evaluations from the design stage is, ultimately, investing in patient safety and business sustainability.

The documentation of these evaluations must reflect not only the quantitative results but also the specific conditions under which they were conducted, with a level of detail that allows for an assessment of the ecological validity of the findings. A report that generally states "it was tested with ten physicians and there were no significant errors" is insufficient and potentially misleading if it does not explicitly specify whether those physicians were well-rested or fatigued, whether they were wearing gloves or not, whether there were scheduled interruptions during the tasks, or whether they were under actual time pressure. To assess their predictive validity for real-world use, competent regulators, and above all the development teams themselves who must learn from experience, need to know the exact conditions under which those results were obtained.

With respect to the physicians and healthcare professionals who participate in these realistic evaluations, the experience can be profoundly revealed at both the personal and professional levels. By seeing themselves make usage errors under time pressure, with induced fatigue, or with constant interruptions, they understand firsthand that these errors are not personal moral failures of carelessness but are predictable consequences of a design that does not adequately account for real-world working conditions. This experiential understanding is valuable not only for improving the specific product they are evaluating but also for changing the organizational culture of individual blame that often prevails in healthcare institutions when incidents occur. When an error is understood as a system or product design problem, the sought-after solution is not to blame and punish the person who acted but to improve the system so that the error is less likely to occur in the future for everyone.

The direct relationship between realistic usability evaluation and patient safety is undeniable and quantifiable. The more closely evaluation conditions resemble actual clinical use conditions, the more likely design problems are to be detected before they cause real harm to people. An

error that occurs during a controlled evaluation is a valuable opportunity for design improvement; an error that occurs in actual clinical practice is an adverse event with potentially serious consequences for health and life. Investing resources in comprehensive, realistic evaluations during development is, therefore, a direct investment in preventive safety. It is not an unnecessary operational expense; it is one of the most cost-effective investments a serious medical software manufacturer can make because it prevents costs associated with incidents, product recalls, litigation, and, above all, harm to people who trusted the product.

5.4 COGNITIVE LOAD AS A DESIGN METRIC

Imagine that a primary care physician must enter a patient's data into an electronic health record system during a fifteen-minute consultation. The screen displays dozens of input fields, some mandatory and marked in red, others optional and of dubious utility, spread across multiple tabs that must be navigated sequentially. The doctor must simultaneously recall what relevant information was requested during the clinical interview, search for it in their hastily written handwritten notes, identify which specific field each piece of data goes into, and meticulously ensure that they do not skip any mandatory steps that would block the entry. Moreover, the patient waits seated, starting at the doctor's back, the clock ticks relentlessly toward the end of the appointment, and the waiting list in the hallway grows with other impatient patients. At the end of a twelve-hour shift, the doctor has spent hours on this bureaucratic task and leaves work mentally exhausted, with the frustrating feeling of having spent more time struggling with the computer than actually listening to and caring for their patients. This experience, which is unfortunately common in modern healthcare systems, is a tangible manifestation of a concept that medical software designers should know and deeply respect: cognitive load.

Cognitive load refers to the amount of mental effort required to perform a specific task using a given interface design. It is not an abstract or metaphorical concept but a physiological reality backed by neuroscience: our brain has a limited capacity to process information simultaneously, especially when it must integrate new information with prior knowledge under pressure. When that capacity is exceeded, cognitive performance decreases, errors of omission and commission increase exponentially, and mental fatigue builds up rapidly.

In clinical settings, where decisions have serious consequences for people's lives and interruptions are constant, managing a professional's cognitive load is not a secondary matter of comfort or job satisfaction but a fundamental aspect of patient safety. A medical product that imposes an excessive cognitive load on its users is not merely annoying or inefficient; it is actively dangerous because it predisposes users to errors at critical moments.

5.4.1 THE THREE TYPES OF LOAD AND THE PRINCIPLES OF REDUCTION

The theory of cognitive load, developed by the educational psychologist John Sweller and subsequently applied to interface design, distinguishes three fundamental types of load that designers must understand and manage. Intrinsic load is that which is inevitably imposed by the complex task itself—that, by its nature, requires mental processing and cannot be eliminated without changing the task. A diagnosis of a rare disease with contradictory symptoms has a high intrinsic load because it requires the integration of a great deal of contradictory medical information; we cannot reduce this load without artificially simplifying the diagnosis. The relevant or germane load is that imposed by the way the task is presented but which helps to perform it better or learn from it. A well-designed flowchart can increase the relevant load because its elements must be processed visually, but in return, it greatly facilitates understanding of the diagnostic algorithm. The irrelevant or extraneous load is imposed by the design without providing any functional benefit, such as having to search for critical data in a deep and confusing menu or remembering a numerical code that should be visible on the screen.

The goal of good medical software design is to drastically minimize extraneous load, optimize relevant load to facilitate learning and accuracy, and scrupulously respect the limits of the intrinsic load without adding unnecessary weight.

For medical software designers, understanding and applying cognitive load theory involves continually asking themselves throughout the design process: What information does the user have to remember at this exact moment in the task? What critical information do they need to keep in mind simultaneously to make a safe decision? What visual, or functional, distractions are

we imposing on them that add no value? Are we making efficient use of their limited processing capacity, or are we squandering it on mechanical tasks with no clinical value? These questions, applied systematically in every design review, reveal opportunities for substantial improvement that often go unnoticed in traditional functional evaluations, which verify only whether the system “works” technically.

One of the fundamental principles for reducing the cognitive load in medical interfaces is recognition over recall. It is neuroscientifically much easier and faster for the human brain to recognize something it is already presented with than to actively recall it from working memory. That is why interfaces that explicitly display available options are cognitively superior to those that force the user to type commands from memory or recall arbitrary numerical codes.

A physician should not have to memorize the ICD-10 codes for the most common diagnoses in their specialty; they should be able to select them from a list well organized by clinical categories with a predictive search. An elderly patient should not have to remember exactly when they took their last dose of medication; the app should clearly display this information on the main screen. This principle, applied consistently throughout the design, drastically reduces the cognitive load and associated memory errors.

Another essential principle is structural and visual consistency. When similar elements behave similarly throughout the system, the user can efficiently transfer their learned knowledge from one part to another without having to relearn new rules for each context. Buttons that activate similar functions should have consistent colors and be in similar positions on the screen. Workflows for analogous clinical tasks should follow analogous navigation structures. Consistency reduces the cognitive load because the brain can automatically apply learned patterns rather than consciously processing each situation as entirely new. In contrast, arbitrary inconsistency forces the user to remain constantly alert to exceptions, which quickly leads to fatigue and increases the likelihood of errors due to incorrect automatism.

5.4.2 VISUAL HIERARCHY, FRAGMENTATION, AND APPROPRIATE METAPHORS

Visual hierarchy is another powerful tool for managing cognitive load in environments where attention is a scarce resource. Not all presented information has the same clinical importance for patient safety and therefore should not compete for the professional's attention with the same visual emphasis. Data critical to immediate safety, such as laboratory values outside dangerous ranges, should stand out visually through size, color, and position, capturing the user's attention even under conditions of fatigue or partial distraction. Secondary supporting data should be available but should not compete for primary attention when more subtle visual coding is used. Data truly irrelevant to the current task should be omitted from the main view or relegated to deep levels of the interface accessible only on demand. A well-designed visual hierarchy guides the user's gaze toward what is important in a preattentive manner, reducing conscious search effort and the risk of overlooking critical information due to information overload.

Intelligent fragmentation of information is another crucial aspect of cognitive load management. Presenting all available information about a patient on a single overwhelming screen overwhelms the user and forces them to perform manual filtering that the design should have done for them through prioritization. Dividing the information into successive screens, each with a clear purpose and a manageable amount of data relevant to that specific purpose, reduces the cognitive load through simplification. However, this principle must be applied with care and moderation: too many successive screens can be just as bad as a single cluttered screen because they force constant navigation and active recall of what was seen on previous screens, increasing the working memory load. The optimal balance lies in grouping information logically, following the natural flow of the clinical task, and always providing a visible navigation “map” that allows the user to know exactly where they are in the process and where they are going.

The use of familiar and consistent metaphors also reduces cognitive load by enabling the transfer of prior knowledge. When a digital interface behaves in a way analogous to something the user already knows from the physical world or other systems, they can apply that existing knowledge without having to learn entirely new associations from scratch. Icons that represent actions through universally recognizable images, approval workflows that mimic traditional paper-based processes, and terms that use the everyday language of the medical profession—all of these facilitate the transfer of prior knowledge and accelerate learning. However, this principle must be applied with critical care: metaphors that do not align well with the software's actual functionality can be more confusing than helpful. A digital trash can that does not allow you

to recover deleted items, a shopping cart that does not display the running total—these are metaphors that mislead and frustrate more than they help.

Immediate and clear feedback is essential for reducing the cognitive load of uncertainty. When the user performs an action on the system, the system should immediately confirm that the action has been received and display the result. Without this feedback, the user wastes valuable cognitive resources, wondering if they did the right thing, if they should repeat the action, or if the system has frozen or is still processing. This active uncertainty affects working memory, which should be devoted to the primary clinical task. A simple progress animation, a brief confirmation message, or a visible status change in the interacting element are sufficient to close the action-feedback loop and relieve the user of that burden of uncertainty.

5.4.3 ERROR PREVENTION, MEASUREMENT, AND SYSTEMIC SUSTAINABILITY

Error prevention through proactive design is among the most effective ways to reduce a system's overall cognitive load. When the system technically prevents the user from making certain common mistakes, it spares them from having to constantly be on guard against those possibilities, freeing up mental resources for other tasks. Input fields that only accept numbers within a valid range, buttons that are visually disabled when prerequisites are not met, and explicit confirmations for critical irreversible actions—all of these act as a safety net that allows the user to focus on what truly matters clinically. However, this principle must be carefully balanced: too many paternalistic restrictions can frustrate and slow things down, especially for expert users who need efficiency and flexibility in unforeseen situations. The design must strike the right balance where error prevention does not become an insurmountable barrier to the necessary clinical efficiency.

For development teams, incorporating cognitive load as an explicit design metric requires specific evaluation methods that go beyond functionality. There are validated standardized questionnaires, such as the NASA-TLX (task load index), that allow users to subjectively assess their perceived cognitive load across different dimensions: mental demand, physical demand, time demand, overall effort, frustration, and perceived performance.⁽²⁾ There are ethnographic observation techniques that identify moments of uncertainty through unexpected long pauses, facial expressions of frustration, or frequent consultations with colleagues.

There are even objective physiological measures, such as pupil dilation or heart rate variability, which can indicate high levels of cognitive effort in real time during interactions. None of these techniques are perfect on their own, but when combined, they provide a rich, multidimensional picture of how a specific design affects users' mental load in practice.

The greater the cognitive load imposed by the design, the greater the statistical probability of human error due to attentional overload. That is why agencies such as the FDA and the EMA are paying increasing attention to this aspect in required usability evaluations. IEC 62366, “(3)”, although it does not explicitly mention the term “cognitive load” in its text, requires a usability engineering process that, when properly applied with attention to these principles, inevitably leads to its reduction. Manufacturers who deliberately ignore this aspect expose themselves not only to less competitive products in the market but also to clinical incidents that could have been avoided with a design more respectful of the limits of the human mind.

For physicians and healthcare professionals who use these products daily, excessive cognitive load manifests tangibly in the form of accumulated mental fatigue, chronic frustration, and, in tragic instances, clinical errors with consequences. When a system forces them to remember too much arbitrary information, to search for critical information poorly organized in deep structures, or to navigate mazes of confusing screens, they are paying a cognitive price they should not have to pay—one that distracts them from their true role of care. That price translates into less quality time for patients, more accumulated work-related stress, and a higher risk of errors during times of peak pressure. Healthcare professionals should be the first to systematically demand that the systems they use respect the limits of their cognitive capacity, and they should actively participate in evaluations that measure this specific aspect by contributing their real-world usage experience.

Finally, excessive cognitive load is not merely an individual problem for the professional experiencing it but also a systemic problem that affects the sustainability of the entire healthcare system. A technological system that chronically imposes a high cognitive load on its users not only affects their immediate performance but also actively contributes to professional burnout, burnout syndrome, and eventually the decision to leave the healthcare profession due to exhaustion. In times of healthcare staffing crises in many developed countries, where the demand for care is growing and the supply of professionals is stagnating, designing technological tools that respect

and care for the minds of those who care for others is not just a matter of product quality but of the sustainability of the public health system. Every conscious effort to reduce unnecessary cognitive load is an investment in the mental health of professionals and, through their well-being and continued presence in the profession, in the health of all potential patients.

Once we have learned to design with users, their limitations, and the real-world context of use in mind, how do we integrate all this knowledge into a systematic process that can be audited and continuously improved?

REFERENCES

What Is IEC 62366? Usability Engineering Guide [Internet]. [cited May 13, 2026]. Available from: <https://www.jamasoftware.com/requirements-management-guide/medical-devices/iec-62366/>

NASA Task Load Index (TLX) - NASA [Internet]. [cited May 13, 2026]. Available from: <https://www.nasa.gov/human-systems-integration-division/nasa-task-load-index-tlx/>

Applying Human Factors and Usability Engineering to Medical Devices | FDA [Internet]. [cited May 13, 2026]. Available at: <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/applying-human-factors-and-usability-engineering-medical-devices>