

Chapter 19

Evaluation of eHealth System Usability and Safety

Morgan Price, Jens Weber, Paule Bellwood, Simon Diemert, Ryan Habibi

19.1 Introduction

Usability and safety are two types of non-functional requirements¹ or quality attributes of a system. Both are increasingly important in health information and communication technology (ICT) systems as they become more integrated into care processes from primary care to the intensive care unit (ICU). Usability and safety are emergent properties of systems, not a property of any particular device such as a piece of computer software. Thus, both should be considered in the context of the *sociotechnical system* of which they are parts. In this chapter, we consider both usability and safety, as we feel they can and should be related.

19.2 Definitions

Sociotechnical systems comprise technology (software and hardware), actors (such as patients, providers, caregivers, friends, and administrators), physical spaces, and the policies that interact, in our case, to support health and wellness. A sociotechnical system in primary care may be a complex web of actors which make up a patient's circle of care and related technologies. For example: a physician office with physicians, nurses, staff and an electronic record; a pharmacy

¹ Non-functional requirements are requirements that do not describe a specific behaviour of a system but rather a requirement that describes how a system is judged to be and is architected into the system as a whole. There are several types of non-functional requirements including: usability, safety, availability, scalability, effectiveness, and testability.

with pharmacists and pharmacy technicians all working through an information system; a person working with their physical trainer who starts using a pedometer and some mobile Health apps to track weight, activity and diet.

Usability is the ease with which a system can be used by the intended actors to achieve specified goals. It also includes a system's learnability. Usability considers satisfaction, efficiency, effectiveness, and context of use (see ISO standard 9241-11). Usability is deeper than the look and feel of a system or user satisfaction; it also includes how a system works in context to complete work or manage workflows, and how well that fits with the needs of users. Usability includes how easy the system is to learn for users and how quickly users can relearn the tool if it is upgraded or if it is not used for a period of time. Finally, usability can positively or negatively impact safety.

Safety is "freedom from those conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment" (United States Department of Defense, 2012). Devices (or components of devices) are referred to as *safety-critical* if they are essential for the safe operations of systems of which they are a part (i.e., their failure alone could result in death, injury, or loss). Otherwise, devices are referred to as *safety-sensitive* if they contribute to safety-critical functions.

Depending on their respective impacts on safety, devices used in eHealth systems may be subject to different levels of mandatory regulation, evaluation, and certification, which may include pre-market evaluation as well as post-market surveillance (Weber-Jahnke & Mason-Blakley, 2012). In practice, however, the classification with respect to their safety impact of many of the devices used in eHealth systems has been challenging. Regulators have struggled to develop a balanced framework for eHealth system evaluation and control. There are two main reasons for these problems: firstly, eHealth devices such as Electronic Medical Records (EMRs) are often complex aggregates of many diverse functions with different criticality; and secondly, systems these devices are integrated into are highly diverse and variable, and by necessity may not be as expected by the device manufacturer.

There are frequent and subtle interactions between the usability and the safety of eHealth systems (see Figure 19.1), which evaluators need to be aware of. In some cases, there may be trade-offs between these two types of requirements. Safety mechanisms may decrease the perceived usability of a system (e.g., where users are required to click on medication alerts while prescribing). Usability enhancements may decrease the safety of a system (e.g., where users are given the opportunity to skip or automate certain tasks). In other cases, increased usability may actually lead to increased safety (e.g., a clean, uncluttered user interface may reduce cognitive load and help prevent medical errors).

The above considerations emphasize the importance of considering larger systems while designing, modelling, and evaluating eHealth devices where sociotechnical aspects of both usability and safety interact (Borychi & Kushniruk,

2010). Thus, it is important to consider safety and usability and their interactions while evaluating any given system.

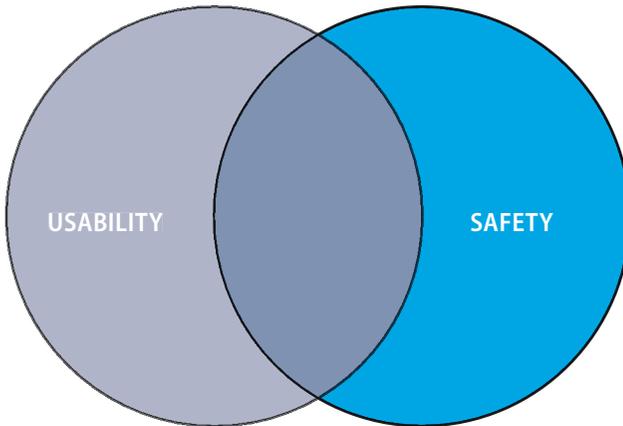


Figure 19.1. Usability and safety requirements often overlap and there is value in considering both.

19.3 When to Evaluate

The importance of evaluating the usability of eHealth systems has been highlighted for almost two decades (Friedman & Wyatt, 1997). Initial usability evaluation in eHealth focused on post-implementation evaluations; however, it has become increasingly evident that these systems should be evaluated sooner in their life cycles, starting from the project planning stages through design and implementation (Kushniruk, 2002; Kushniruk & Patel, 2004; Marcilly, Kushniruk, Beuscart-Zephir, & Borycki, 2015). Conversely, initial safety evaluation efforts of eHealth systems have focused on pre-implementation evaluations, while more recent evidence indicates the insufficiency of this approach and the need for additional post-implementation evaluations.

Ideally, evaluation of usability and safety of eHealth systems should occur throughout their life cycle — during conception, design, development, deployment, adoption, and ongoing evolution. While evaluation should be considered throughout the life cycle, the methods and focus of the evaluation may change over time. Current evaluations of eHealth systems are aimed at evaluating the technology in early stages of design to make informed design decisions and reduce risks; additionally, evaluating during implementation and post-deployment to assess the impact of a system and improve future system revisions (Marcilly et al., 2015). Earlier evaluation during design and/or procurement of systems is considerably less expensive than trying to change existing tools and processes post-implementation.

Choosing not only the proper methods to evaluate eHealth systems throughout their life cycles but being aware of the contexts in which to evaluate these

systems is essential (Kuziemsky & Kushniruk, 2014, 2015). For example, when designing a system, one can employ usability testing and safety inspection methods on low fidelity prototypes and workflow designs, respectively. As a system is deployed, observational studies are very useful to understand how it is used in practice and one may see surprising workflows, workarounds, and unintended consequences. Thus, these different methods help support decision-making with regard to the eHealth system, how it is designed, configured, and implemented.

19.4 Usability Methods

There are many methods for assessing and improving the usability of systems. It is helpful to broadly categorize these methods first, before providing a few examples. Usability methods can be broadly categorized into *inspection methods* and *testing methods*. Usability inspection methods, as a group, are expert-driven assessments of a design or product's usability. They do not involve users. Usability testing methods, by contrast, engage real-world users — potential or expected users — to explore user interfaces, often completing important or common tasks within the system that test both the user interface and user experience.

Both types of usability methods can vary in their focus. For example, they can be very granular, focusing on an individual's interaction with the eHealth application, or they can focus on the broader interactions between actors in a group. Table 19.1 provides some examples in each category. A system's usability can be evaluated in different settings, including real (i.e., in-situ) or simulated environments (i.e., clinical simulations in a usability lab). Using clinical simulations for usability evaluations often results in higher evaluation fidelity (Borycki, Kushniruk, Anderson, & Anderson, 2010; Li et al., 2012).

Table 19.1

Usability Methods Categorized by Type and Focus

	Individual Focus	Group Focus
Inspection	<ul style="list-style-type: none"> • Cognitive Task Analysis • Heuristic Inspection 	<ul style="list-style-type: none"> • Distributed Task Analysis
Testing	<ul style="list-style-type: none"> • Think Aloud User Testing 	<ul style="list-style-type: none"> • Observational Studies • Contextual Analysis

- *Cognitive Task Analysis* is a form of expert inspection that focuses on the cognitive needs of an individual user (in a particular role) as they complete tasks. Cognitive Task Analysis is well suited for eHealth systems; much of healthcare is focused on the cognitively intensive tasks of collection and synthesizing patient information for diagnoses and managing treatment.

- *Think Aloud* is a common form of usability testing where individual users are asked to use an application and encouraged to speak their mind while completing tasks. By thinking aloud in the moment, the designers are able to capture usability challenges that might not otherwise be remembered by the user in follow-up interviews. Multiple users are asked to individually complete a set of tasks in the application, typically while being recorded. The analyst then reviews the session (or their notes) to highlight usability challenges in using the system to complete the tasks. The findings across the multiple test sessions are then synthesized into design recommendations that can be implemented and retested.
- *Distributed Task Analysis* builds on the theory of Distributed Cognition (Hutchins, 1995) and is a model that expands the concept of cognition outside of the mind to groups of actors (both human and technical). Understanding how a patient is kept alive in a trauma in an emergency or during surgery are two examples where a distributed task analysis would be helpful as there are many actors working together in parallel. Like cognitive task analysis, distributed task analysis is an inspection method; however, the scope is typically larger, considering how a process unfolds and how groups of actors (and in this case eHealth tools) work together to come to decisions and complete actions.
- *Observational Studies* place the analyst within an environment to observe the context of work. There are several approaches to observational studies, with varying focus, methods for recording observations (from note taking to digital recording of audio and video), and duration. Observational studies permit better understanding of the interactions between the technology and the interdependent workflow between actors (people, patients, physicians, nurses, etc.). Observations can take place at single or multiple locations and may focus on care flows of single patients through the healthcare system, or can be team focused, observing how a ward or department might work.

19.5 Safety Methods

As highlighted previously, the quality attribute of safety is often linked to that of usability. Consequently, the usability evaluation methods as characterized above may also be helpful for identifying safety-related concerns, in particular when it comes to safety concerns related to human factors and human-computer interaction. A variety of methods have been developed for evaluating sys-

- 2 *Failure Modes and Effects Analysis (FMEA)* is a method developed by the safety engineering community, which has also been adapted to healthcare as Healthcare FMEA (HFMEA), and has been used by the U.S. Department of Veterans Affairs (DeRosier, Stalhandske, Bagian, & Nudell, 2002). The method is based on a process model describing the relevant workflows within a particular system. It systematically identifies potential failure modes associated with the system's components and determines possible effects of these failures. Failures are assigned criticality scores and are ranked accordingly. Control measures are developed to mitigate accidents that could result from the most critical failure modes. HFMEA can be used early in the design of new systems or processes and also much later as the sociotechnical systems evolve with time and use.
- 3 *Fault Tree Analysis (FTA)* is a deductive method that starts by assuming safety faults and successively seeks to identify conditions under which system components could lead to these faults (Xing & Amari, 2008). An example of a system fault in the healthcare domain could be *patient has an adverse reaction to a medication*. Conditions which could lead to such a fault could include malfunctions of the clinical decision support system (for showing drug allergy alerts), malfunction of the communication system between the EMR and pharmacy, missing or incongruent data in the EMR about the patient (allergies, other active medications, etc.), or other factors. FTA successively analyzes potential causes for safety faults in a hierarchical (tree-like) structure; this is a deductive approach and complementary to FMEA, which is inductive in nature. By contrast, FMEA starts from system components, their potential failure modes and focuses on determining possible faults that could result from them.
- 4 *Hazard and Operability (HAZOP)* is another process-based safety evaluation method, which was originally developed in the design of industrial chemical plants, but has since been used for computer-based systems (Dunjó, Fthenakis, Vilchez, & Arnaldos, 2010). HAZOP relies on a disciplined, systematic process of using guidewords to discover potential unintentional hazardous consequences of process deviations. Typical HAZOP guidewords include “no”, “more”, “less”, “as well as”, “reverse”, etc. These guidewords are applied to actions modelled in the process under investigation to identify possible process deviations and their (potentially safety-relevant) consequences.

19.6 Selected Case Study Examples

The following two examples have been selected because they both have aspects of usability and safety. The first example is primarily safety focused, examining a commonly cited case study of a computer-based physician order entry (CPOE) system. The second example illustrates how usability design standards were developed in order to improve overall safety of eHealth in the United Kingdom's National Health Service (NHS).

19.6.1 Safety Case Study: A Technology-induced Medication Error

The first case study involves a CPOE system deployed at the New York Presbyterian Hospital. Horsky, Kuperman, and Patel (2005) analyzed the factors that led to a technology-induced medical accident, while Weber-Jahnke and Mason-Blakley (2012) provided a further systematic analysis using a STAMP. In this incident, an elderly patient was admitted to the hospital and received a significant overdose of Potassium Chloride (KCl) over a period of two days, involving multiple medication orders by multiple providers. Notably, no single event can be pinpointed as the root cause for the accident and the CPOE device functioned as intended by the manufacturer. Rather, the accident was the result of a number of factors that in combination resulted in the harmful outcome.

The following is a series of significant events leading to the harmful outcome (i.e., an accident):

- 1 On Saturday, Provider A reviews the results of a lab test and finds the patient hypokalemic (deficient in bloodstream potassium).
- 2 Provider A orders a KCl bolus injection using the CPOE.
 - a. Provider A notices that the patient has an existing drip line and decides to use the line instead of an injection.
 - b. Provider A enters a new drip line order and intends to cancel the injection order.
 - c. However, Provider A inadvertently cancels a different (outdated) injection order, which had been entered by a different provider two days prior.
- 3 Provider A is notified by the pharmacy because the dose for the drip order exceeds the hospital's maximum dose policy.
- 4 Provider A enters a new drip order but fails to enter it correctly (a maximum volume of 1L was entered but in the wrong input field, namely the "comment" field).

- a. Provider A enters this information in the “comment” field as free text but fails to enter it in the structured part of the CPOE input form.
- 5 The KCl fluid continues to be administered for 36 hours, in addition to the initial bolus injection that ran to completion.
- 6 On Sunday morning, Provider B takes over the case and checks the patient’s KCl level based on the most recent lab test (which was still from Saturday).
- 7 Not realizing that the patient’s initial hypokalemic state had already been acted upon, Provider B orders two additional KCl injections.
- 8 On Monday morning a KCl laboratory test found the patient to be severely hyperkalemic. The patient was treated immediately for hyperkalemia.

This case study highlights several aspects related to usability, safety, and the interaction between these two system quality attributes:

- A. The failure to specify an effective stop date / maximum volume for Provider A’s drip order is a direct result of a usability problem. The CPOE input form allowed the provider to make free text comments on the order, but these comments were not seen as instructions by the medication-administering nurses.
- B. The failure of Provider B to realize the patient’s hypokalemic state is a clear system (safety) design problem. The device could have been designed to relate ordered interventions to out-of-range test results, and make providers aware of the fact that test results had already been acted on.
- C. The failure of Provider A to cancel the right order cannot clearly be categorized as a sole usability or safety problem, respectively. Rather, it relates to both aspects. On one hand, the device could have made it easier to distinguish old (and new) orders from orders submitted by other providers (and in the past). On the other hand, a more effective design of the CPOE device could have detected an overdose violation based on the consideration of multiple orders rather than based only on the consideration of each order separately.

Usability and safety evaluation studies may have prevented or mitigated the above accident. For example, Think Aloud user testing with providers may have indicated that providers tend to use the “comment” field of the CPOE device to specify volume limits, while administering nurses would disregard that field (see point A above). Safety evaluation methods may have prevented point B. For example, the application of HAZOP guidewords like “as well as” on the order entry process step (after the lab review step) may have revealed the hazard of prescribing interventions more than once as a reaction to a specific lab test. Ideally, proper design mitigation would have flagged the out-of-range lab test as “already acted upon” in the EMR. Finally, usability or safety evaluation methods could have mitigated point C above. For example, cancelling the wrong medication order is a clear failure mode of the ordering system (FMEA), which could be mitigated by checking whether the cancelled order is current, or has already been administered in the past. Moreover, HAZOP guidewords could have identified the hazard of medication overdoses due to two or more concurrent medication orders of the same substance.

19.6.2 Usability Case Study: Common User Interface

The *Common User Interface* project (CUI) was an attempt to create a safer and more usable eHealth user interface by defining a standard across multiple clinical information systems that would be consistent for users. This project was undertaken as a joint effort between the U.K.’s National Health Service (NHS) and Microsoft. Safety through improved user interface design was a key consideration. As part of a larger project, CUI set about to create *design guidances* that presented a standard (common) user interface approach for aspects of eHealth tools that would better support care. Further, this would support clinicians who were moving between different eHealth systems. The CUI design guidances were published and cover a range of topics within the following:

- Patient identification
- Medications management
- Clinical notes
- Terminology
- Navigation
- Abbreviation
- Decision support

Each design guidance is an extensive document that addresses a component of one of the topics above. For example, as part of the medications management guidelines, there are detailed documents for “drug administration”, “medication line”, and “medication list” among others that help developers with specific information on how to (and how not to) implement the user interface. The design guidance documents were developed in a manner compliant with the Clinical Safety Management System defined by the NHS. Furthermore, the guidelines include the rationale for the recommendations (and associated evidence).

For example, the medication line design guideline (v2.o.o)² carefully describes how a medication should be displayed. It includes specific recommendations for display of generic names, brand names, strength, dose, route, and frequency. These include rationale for font styles, spacing, and units that make information easier to read, to comprehend, and reduce the risk for misinterpretation. Figure 19.3 demonstrates CUI guidances such as: “generic medication name must be displayed in bold”; “dose must be clearly labelled”; “acronyms should not be used when displaying the medication instructions”; and “instructions should not be truncated but all instructions must be shown, with wrapping if necessary” (note oxycodone uses three lines).

Current Medications

oxycodone - OXYCONTIN -
modified release tablet - **DOSE 10 mg** - oral - every
twelve hours

metronidazole - FLAGYL - tablet - **DOSE 500 mg** -
oral - twice a day

Figure 19.3. An example of medication display following CUI design guidance.

The Microsoft Health Patient Journey Demonstrator was built to demonstrate how CUI guidances could be implemented on a Microsoft platform to display health information in a health information system (Disse, 2008). This example, showing how CUI could be applied to primary care, secondary care, as well as administrative clinical interfaces, has attracted attention from various communities due to its applicability and as a solution to provide a standardized approach to clinical user interfaces. The CUI design guidances are freely available³. Microsoft® also provides some free example software controls under the Microsoft Public License.

CUI was an impressive effort and reviewing many of the guidelines in these design guidances provides a wealth of information on how to and how not to

2 <http://systems.hscic.gov.uk/data/cui/uig/medline.pdf>

3 <http://systems.hscic.gov.uk/data/cui/uig>

design user interfaces in the health domain. However, CUI only covered a small number of areas and the project has not continued. The knowledge that was generated is freely available at mscui.org and through the NHS.

19.7 Summary

Usability and Safety are increasingly being acknowledged as necessary components for the success of eHealth. However, achieving safe and usable systems remains challenging. This may be because it is often unclear how to measure these quality attributes. Further, as systems are deployed and adopted, it becomes harder and more costly to make large changes. This is especially the case as eHealth tools are being increasingly integrated into care processes across the circle of care, and as people and providers use an increasing range of tools, apps and health records to manage care.

A single, large “safety review” or “usability inspection” is less likely to have a long-lasting impact. Instead organizations should focus on embedding usability and safety in their culture and process. Thus, we encourage that safety and usability engineering should occur throughout the life cycle of eHealth tools from requirements and procurement to ongoing evaluation and improvement. In this chapter we have highlighted a few methods for evaluating safety and usability. It is likely more feasible to build on existing work, such as the CUI project, and use multiple methods to triangulate findings across small evaluation projects than it is to attempt a large, comprehensive study with a single method; multiple methods complement each other.

Policy-makers, funding programs, and health organizations should explicitly embed safety and usability engineering into the operational eHealth processes. There is increasing need for both usability and safety engineers in health as eHealth systems are being, and continue to become, broadly adopted.

References

- Borycki, E., & Kushniruk, A. (2010). Towards an integrative cognitive-socio-technical approach in health informatics: Analyzing technology-induced error involving health information systems to improve patient safety. *The Open Medical Informatics Journal*, 4, 181–187.
doi: 10.2174/1874431101004010181
- Borycki, E., Kushniruk, A. W., Anderson, J., & Anderson, M. (2010). *Designing and integrating clinical and computer-based simulations in health informatics: From real-world to virtual reality*. Vukovar, Croatia: In-Tech.
- DeRosier, J., Stalhandske, E., Bagian, J. P., & Nudell, T. (2002). Using health care failure mode and effect analysis: The VA National Center for Patient

- Safety's prospective risk analysis system. *The Joint Commission Journal on Quality Improvement*, 28(5), 248–267.
- Disse, K. (2008). Microsoft health patient journey demonstrator. *Informatics in Primary Care*, 16(4), 297–302.
- Dunjó, J., Fthenakis, V., Vílchez, J. A., & Arnaldos, J. (2010). Hazard and operability (HAZOP) analysis. A literature review. *Journal of Hazardous Materials*, 173(1), 19–32. doi: 10.1016/j.jhazmat.2009.08.076
- Friedman, C. P., & Wyatt, J. (1997). *Evaluation methods in medical informatics*. New York: Springer.
- Horsky, J., Kuperman, G. J., & Patel, V. L. (2005). Comprehensive analysis of a medication dosing error related to CPOE. *Journal of the American Medical Informatics Association*, 12(4), 377–382. doi: 10.1197/jamia.M1740
- Hutchins, E. (1995). How a cockpit remembers its speeds. *Cognitive Science*, 19(3), 265–288.
- Kushniruk, A. (2002). Evaluation in the design of health information systems: Application of approaches emerging from usability engineering. *Computers in Biology and Medicine*, 32(3), 141–149. doi: 10.1016/S0010-4825(02)00011-2
- Kushniruk, A. W., & Patel, V. L. (2004). Cognitive and usability engineering methods for the evaluation of clinical information systems. *Journal of Biomedical Informatics*, 37(1), 56–76. doi: 10.1016/j.jbi.2004.01.003
- Kuziemsky, C. E., & Kushniruk, A. (2014). Context mediated usability testing. *Studies in Health Technology and Informatics*, 205, 905–909.
- Kuziemsky, C., & Kushniruk, A. (2015). A framework for contextual design and evaluation of health information technology. *Studies in Health Technology and Informatics*, 210, 20–24.
- Leveson, N. (2012). *Engineering a safer world: Systems thinking applied to safety*. Cambridge, MA: MIT Press.
- Li, A. C., Kannry, J. L., Kushniruk, A., Chrimes, D., McGinn, T. G., Edonyabo, D., & Mann, D. M. (2012). Integrating usability testing and think-aloud protocol analysis with “near-live” clinical simulations in evaluating clinical decision support. *International Journal of Medical Informatics*, 81(11), 761–772. doi: 10.1016/j.ijmedinf.2012.02.009

- Marcilly, R., Kushniruk, A. W., Beuscart-Zephir, M., & Borycki, E. M. (2015). Insights and limits of usability evaluation methods along the health information technology lifecycle. *Studies in Health Technology and Informatics*, 210, 115–119.
- United States Department of Defense. (2012). Standard practice for system safety: MIL-STD-882E. Retrieved from <http://www.system-safety.org/Documents/MIL-STD-882E.pdf>
- Weber-Jahnke, J., & Mason-Blakley, F. (2012). On the safety of electronic medical records. In Z. Liu & A. Wassing (Eds.), *Foundations of Health Informatics Engineering and Systems: First international symposium, FHIES 2011* (pp. 177–194). Berlin: Springer.
- Xing, L., & Amari, S. V. (2008). Fault tree analysis. In K. B. Misra (Ed.), *Handbook of performability engineering* (pp. 595–620). London: Springer.