# The Unintended Effects of Medical Software on Clinical Decisions and Patient Safety: A System Viewpoint

Farah Elkourdi
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, NJ 07030
e-mail: felkourd@stevens.edu

Onur Asan
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, NJ 07030
e-mail: oasan@stevens.edu

Mo Mansouri
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, NJ 07030
e-mail: mo.mansouri@stevens.edu

***Abstract*—Integrating medical software into the healthcare system began as a way to reduce medical errors and simplify procedures. However, medical software-related errors have been a source of concern for both physicians and patients. Software-related medical errors have affected clinical decisions, which threaten patient safety. This paper explores the potential sources of unintended negative consequences associated with medical software. We used systems thinking methods to explore the relationship between software-related errors and their overall impact. We also developed insights into how to improve patient safety by eliminating software-related medical errors system-wide.**

***Keywords- Medical Software; Healthcare; System Thinking; Clinical Decisions; Patient Safety; Medical Errors; Human Factors.***

## I. INTRODUCTION

According to the World Health Organization, patient safety is a healthcare discipline that emerged with the evolving complexity of healthcare systems and the resulting rise of patient harm in healthcare facilities, which aims to prevent and reduce risks, errors, and harm that occur to patients during the provision of health care [1]. Patients cannot always notice the consequence of medical software, although it is a vital component of the healthcare system impacting patient safety. Medical software was integrated into healthcare to assist physicians in areas including managing patients' data, improving care coordination, and proper diagnosis. There are several types of medical software used by physicians, such as Electronic Health Records (EHR), E-prescriptions, and Medical diagnosis software. Medical software can enhance the quality of patient care, diminish paper workload, and reduce unnecessary medical tests [2].

Over time, the healthcare system gradually shifted toward integrating medical software. It started in the 1960s when beliefs began to arise around computer technology holding promises for improved decision-making by clinicians [3]. Software development started to evolve over years with noticeable expansion in the healthcare industry. In 2003, both the public and private sectors took major steps to ensure that EHRs would be a component of medical offices within five to eight years [3]. Nowadays, medical software is adapted and widely used by physicians in the healthcare field.

However, medical errors are a serious public health problem and the third-leading cause of death [4]. There are problems associated with healthcare Information Technology (IT) that can disrupt care delivery and harm patients [5]. From this standpoint, what factors have led medical software to become a source of harm despite its primary objective to improve patient safety? Systems and Software Engineering contribute to the advancement and improvement of healthcare delivery and its safety [6]. In this paper, we will analyze, from a system viewpoint, the reasons behind causing the unintended consequences of medical software on clinical decisions and patient safety. A human being usually reacts to immediate circumstances, but it is more difficult to analyze how our previous actions and decisions may influence a specific situation in the future [7]. We will use system thinking methods to identify the relationship between events and potential improvements.

The rest of this paper is organized as follows. Section II describes the potential sources of software medical errors in healthcare. Section III describes the stakeholders' perspectives on medical software development. Section IV addresses some of the unintended consequences related to medical software that affect clinical decisions, patient care, and safety. Section V discusses the conclusion of integrating medical software into healthcare systems.

## II. POTENTIAL SOURCES OF UNINTENDED CONSEQUENCES

Medical software is a computer program intended for use in the diagnosis, cure, mitigation, treatment, or prevention of disease, or intended to affect the structure or any function of the body of humans; or used in the production or control of drugs, devices, data, or other uses in medical diagnosis or treatment [8]. Implementing software into the healthcare system might produce possible unforeseen consequences due to the complexity of the healthcare system and various stakeholders, such as the development team, policymakers, and physicians. The successful implementation of medical software involves assessing multiple aspects, such as its interaction with the people, processes, and culture of the

 ISBN: 978-1-68558-037-7 

organization to deliver safe care for patients and a more satisfying work experience for clinicians and staff [9].

Figure 1 shows the potential sources of unintended consequences related to medical software in the healthcare system. Eliminating the negative consequences of medical software and identifying their potential sources requires understanding the subsystems' interaction. IT development teams should concentrate on developing reliable software, especially for critical industries, such as the healthcare field. The development of reliable software consists of a series of phases that shape the software system's complexity. The phases start with requirements and end when the product reaches the market in addition to applying verification and validation techniques throughout the different phases [10]. The development process involves difficulties in managing the medical software without negatively affecting patient safety, privacy, and security. As a result, regulations were enacted and evolved for medical software development to ensure patient safety and protect patients' data. For example, U.S. Food and Drug Administration (FDA) intends to apply its regulatory oversight to the medical device software functions whose functionality could pose a risk to a patient's safety if the device were not to function as intended [11]. Besides, organizations that use medical records have to follow privacy and security regulations for managing patient information in the healthcare industry [12]. The impact of regulations and policies on medical software is presented in green arrows in Figure 1.

Medical software risks can arise from flaws in the software itself, introduced during development, unintended consequences, or the physician's use of the software [13]. Software medical errors caused in a hospital environment involve workload, stress, and fatigue integrated with usability issues, such as poor software interface design, and reliability issues, such as functional errors [4][5][14]. Software quality attributes are the nonfunctional requirements that define the software system from the user's point of view, and it represents a critical component of the software system to achieve the field objectives. Medical software quality attributes include reliability, usability, safety, privacy, and security. In healthcare, poor software quality may have safety implications; for example, using an electronic prescribing system that corrupts data about the dose of a prescription, or a patient management system that stores incorrect information about the criticality of the condition [13]. The blue arrows in Figure 1 present the impact of medical software quality on the healthcare environment.

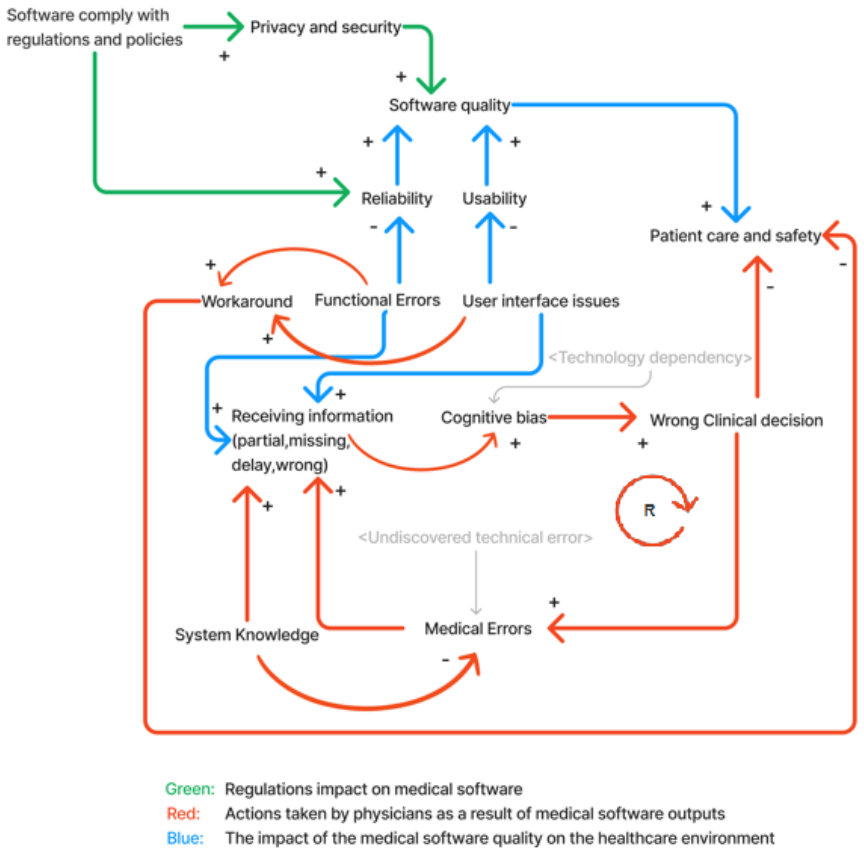


Figure 1. Causal loop presents the potential sources of software medical errors

 ISBN: 978-1-68558-037-7 

The healthcare field is a dynamic and complex environment. The physicians might be under psychological (anxiety, grief, and guilt) and cognitive (compassion dissatisfaction, burnout, and stress) pressure [4]. When developers make mistakes, the consequences may not be foreseen for a long time; the physicians have to respond rapidly although they are under substantial workload pressures, and they may not understand that their problems are triggered by faulty code [15]. Technical issues could result in physicians receiving wrong, incomplete, delayed, or missing information [5]. Physicians might use workarounds to overcome technical issues. Workaround caused the system to be used in a way that it is not intended to be used which might cause medical errors [14]. Hazards might turn into risks causing harm generated by the physicians due to the limited amount of knowledge known about the software system. Knowledge refers to understanding the system's functionalities. Physicians might not use some functions due to a lack of knowledge and proper training, which could reduce the productivity and efficiency of the healthcare system [14]. The workflows in the healthcare environment have complexity in task structures as physicians manage multiple competing demands under resource constraints, and from psychology, the complexity leads to cognitive load and error [16]. Besides, human cognition, its limitations, and reliance on heuristics can affect human decision-making processes, which can alter how humans weigh the importance of data when making a decision [16]. Unexpected errors will always be a part of the medical system due to the universal nature of human fallibility and technology [4]. The red arrows in Figure 1 present physicians' actions as a result of medical software outputs. The key objective is to eliminate the potential sources of the negative consequences to deliver a high care quality in clinical decisions and support patient safety.

## III. Different Perspectives and Values

In Figure 2, we demonstrate the different perspectives of multiple stakeholders in the system. The four perspectives are human factors in developing software that adapts to the healthcare field, meeting deadlines for business market races and competitions, considering the technical aspect to develop stable software, and applying regulations to protect patients' safety, data privacy, and security. The ideal point for producing medical software is to maintain in the middle to observe the full picture that is balancing between all four different values without drifting to one aspect. For example, considering the software functionality technical aspect from one perspective will result in losing other values, such as human factors from the physicians' perspective. The primary objective should be the patients' right to provide safe, reliable, and patient-centered care [4].

### A. *Policymakers' perspective*

Policymakers have established regulations to ensure the safety and privacy of patient data. Developers must consider government regulations, human factors, and stable functionalities while developing medical software. Health Insurance Portability and Accountability Act (HIPAA) is a federal law that protects sensitive patient health information from being revealed without the patient's permission or knowledge [17]. For example, electronic health record systems are required to have audit functions to detect the identity of the users accessing the system [18]. The violations of regulations may not only cause the disclosure of patients' sensitive information, but also can cause no-tolerance penalties and termination to the healthcare providers [18].

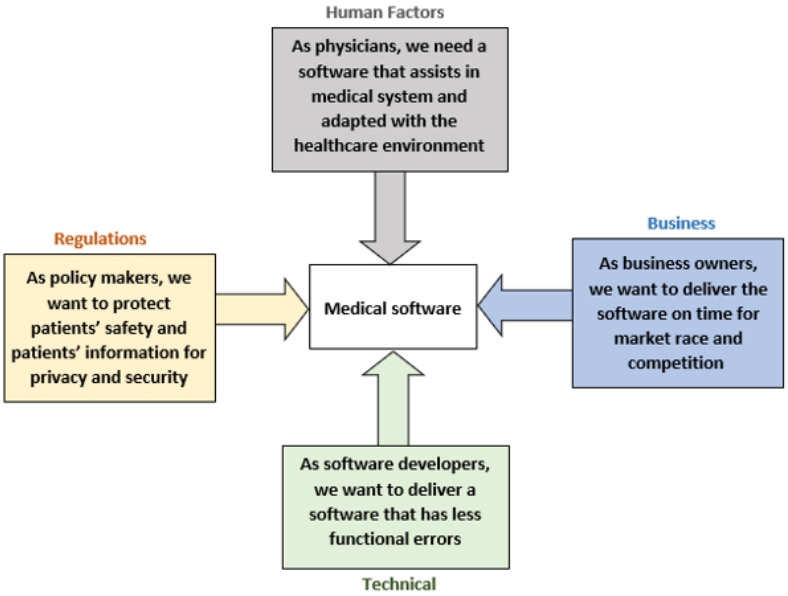


Figure 2. Different perspectives of multiple stakeholders in the system

### B. *Technical versus business perspective*

Software developers are concerned with avoiding functional errors. As a result of the industry's race to market [19], software developers may be under pressure to meet deadlines and market competition, which might cause spaghetti code that is more prone to errors. Spaghetti code is unstructured code programmed by software developers and can be caused by taking shortcuts to meet deadlines. Delivering unstructured code might result in a chain of errors that will appear when physicians are using the system. More errors will start to appear while fixing one error due to the spaghetti unstructured code [19]. This concludes to a complex software system that is difficult to track, maintain or extend and accordingly results in a prone code to errors. Healthcare organizations must collaborate with their software vendors to monitor and optimize the used technology to help them identify, measure, and improve the quality and safety of the care provided [20].

### C. *Human factors perspective*

Physicians focus on having medical software that would allow for the best patient care and adapted to their workflow in healthcare. Medical errors may occur if the medical software development team fails to consider human factors perspectives when developing a stable software system for a healthcare environment. For example, developing a drop-down menu listing 86 options that are irrelevant for a specified patient might cause physicians to make errors by clicking on the wrong dose or form [19]. This could have an impact on clinical decisions and patient safety. In this case, the development team did not deliver a functional error but missed the other values from different perspectives.

Human factors for medical software involve designing user interfaces so that physicians can complete their tasks without making errors that might affect patient safety. There are some approaches to include human factors in user interface development, such as hiring human factors engineers or psychologists directly into development teams, placing a development group under the leadership of a human factors professional, or forming an educational center in which software engineers learn about human factors approaches [21]. Effective human factors methods could be applied routinely and during the software development life cycle phases, which could minimize the negative unintended consequences and reduce the chance of errors [22].

Errors and poor user interfaces might interfere with receiving the information and lead to errors when making decisions [5]. To address patient safety that arises from human error as well as other sources, systems, and software engineering attention must increasingly focus on continuously creating robust, reliable, and dependable applications and infrastructure focused on addressing needs at the point of delivery of care [6]. Medical software containing software functional errors or design issues is causing hazards in the healthcare environment. Suppose the physician did not discover the functional errors or the design issues in the early stage. In that case, it will turn into a risk that will negatively affect the clinical decisions and patient safety, as shown in Figure 3. If the physician detects the error in the early stage before it turns into a medical error, it will cause a workload to work around the error. However, it will still be a hazard caused by the workarounds that might affect patients' safety, as mentioned in Figure 3.

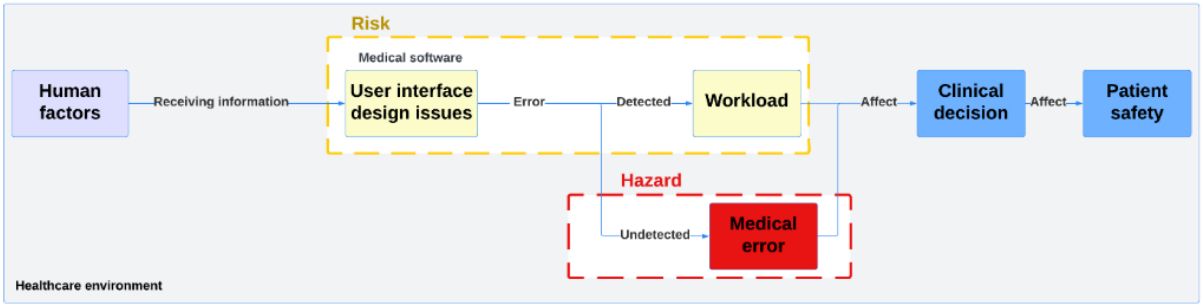


Figure 3. The impact of human factors and medical software on clinical decision and patient safety

## IV. SOME UNINTENDED CONSEQUENCES RELATED TO MEDICAL SOFTWARE AFFECT CLINICAL DECISIONS, PATIENT CARE, AND SAFETY

According to the "potential sources of unintended consequences" section, some unintended consequences caused by medical software affect the clinical decision and patient safety, such as causing new types of errors in healthcare, producing new/more workload on physicians, and technology dependency in taking decisions. Commission errors are the most commonly reported software medical errors in healthcare generated by wrong data entry, selection from dropdown menus, and file uploads [5]. The effects of it cause errors and delays in clinical decision including medication administration errors and failure to follow up test results [5].

Medical software may increase the hazards in the healthcare field and cause new types of errors due to the doctor's computer interaction or functional errors generated during the software development. Software developers must optimize the design of human-computer interfaces because interface design issues cause many medical errors [23]. The dependency on medical software is one of the consequences that could have a negative impact. If the system is down, physicians should ensure that basic medical care can continuously be provided in the absence of technology [18]. Furthermore, medical software could generate new or more work for physicians as unintended consequences. Accordingly, developers should enhance the user interface to reduce the collection of redundant information, display relevant information in logical locations and reduce the amount of typing [23].

## V. CONCLUSION

As healthcare systems continue to rely more heavily on medical software, it is crucial to evaluate their effectiveness and safety. In this paper, we discuss the potential sources of negative consequences generated by integrating medical software into the healthcare field from a systems perspective. Our discussion reveals that the challenges rely on applying a complex system "the medical software" into another complex system "the healthcare" system. We started with the primary reasons for integrating medical software in the healthcare field and its evolution over the years. We demonstrate the interaction between software developers, policymakers, and healthcare providers. Besides, introducing the different values and perspectives in delivering medical software. Our objective is to identify potential sources of unintended consequences in order to eliminate negative outcomes.

The system stakeholders must collaborate and communicate effectively to ensure that the medical software is developed and implemented to prioritize patient safety and care. This requires collaboration between healthcare providers and software developers to provide continuous user feedback and develop a user-centered design. Consequently, medical software can effectively support healthcare providers in delivering and improving outcomes.

## REFERENCES


[1] W. H. Organization, "Patient Safety," World Health Organization, 13 September 2019. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/patient-safety. [Accessed 21 March 2023].

[2] healthit.gov, "Advantages of Electronic Health Records," healthit.gov, 8 March 2022. [Online]. Available: https://www.healthit.gov/faq/what-are-advantages-electronic-health-records. [Accessed 21 March 2023].

[3] E. Ambinder, "A History of the Shift Toward Full Computerization of Medicine," American Society of Clinical Oncolog, vol. 1, no. 2, pp. 54-56, 2005.

[4] O. Ozeke, V. Ozeke, O. Coskun, and I. I. Budakoglu, "Second victims in health care: current perspectives," Advances in Medical Education and Practice, vol. 10, pp. 593–603, 2019.

[5] M. O. Kim, E. Coiera, and F. Magrabi, "Problems with health information technology and their effects on care delivery and patient outcomes: a systematic review," Journal of the American Medical Informatics Association, vol. 24, no. 2, pp. 246-250, 2017.

[6] R. A. Schrenker, "Software engineering for future healthcare and clinical systems," Computer, vol. 39, no. 4, pp. 26 - 32, 2006.

[7] H. Sibo-Ingrid, G. Celis-David, and S. Liou, "Systems Thinking: a paradigm for advancing technology to enhance humanity," in IEEE International Conference on Orange Technologies (ICOT), Nusa Dua, Bali, 2018, pp. 1-6, doi: 10.1109/ICOT.2018.8705910.

[8] P. Schneider and M. Hines, "Classification of medical software," in Symposium on Applied Computing, Arkansas, 1990, pp. 20-27, doi: 10.1109/SOAC.1990.82134.

[9] Healthit.gov, "Implementing Health IT," [Online]. Available: https://www.healthit.gov/topic/safety/implementing-health-it. [Accessed 21 March 2023].

[10] D. Leggingwell and B. Norman, "Software quality in medical devices-a top-down approach," in Computer-Based Medical Systems-Proceedings of the Sixth Annual IEEE Symposium, Michigan, 1993, pp. 307-311, doi: 10.1109/CBMS.1993.263001.

[11] FDA, "Policy for Device Software Functions and Mobile Medical Applications," U.S. Food & Drug, September 2022. [Online]. Available: https://www.fda.gov/regulatory-information/search-fda-guidance-documents/policy-device-software-functions-and-mobile-medical-applications [Accessed 21 March 2023].

[12] D. Baumer, J. B. Earp, and F. C. Payton, "Privacy of medical records: IT implications of HIPAA," ACM SIGCAS Computers and Society, vol. 30, no. 4, pp. 40–47, 2000.

[13] G. Despotou, S. White, T. Kelly, and M. Ryan, "Introducing safety cases for health IT," in ICSE Workshop on Software Engineering in Health Care, SEHC, Zurich, 2012, pp. 44-50, doi: 10.1109/SEHC.2012.6227010.

[14] L. Salahuddin and Z. Ismail, "Antecedents for safety in health IT: An exploratory investigation," in Proceedings of


 ISBN: 978-1-68558-037-7 

the 6th International Conference on Information Technology and Multimedia, Putrajaya, 2014, pp. 38-43, doi: 10.1109/ICIMU.2014.7066600.

[15] H. Thimbleby, "Inside Medical Software: When Programming Errors Cost Lives," ITNow, vol. 60, no. 2, pp. 50-52, 2018.

[16] E. Coiera, J. Ash, and M. Berg, "The Unintended Consequences of Health Information Technology Revisited," Yearb Med Inform, vol. 25, pp. 163–169, 2016.

[17] CDC, "Health Insurance Portability and Accountability Act of 1996 (HIPAA)," Centers for Disease Control and Prevention, 27 June 2022. [Online]. Available: https://www.cdc.gov/phlp/publications/topic/hipaa.html#:~:text=The%20Health%20Insurance%20Portability%20and,the%20patient's%20consent%20or%20knowledge [Accessed 21 March 2023].

[18] N. Menachemi and T. H. Collum, "Benefits and drawbacks of electronic health record systems," Risk Manag Healthc Policy, vol. 4, pp. 47-55, 2011.

[19] F. Schulte and E. Fry "Death By 1,000 Clicks: Where Electronic Health Records Went Wrong," KHN, 18 MARCH 2019. [Online]. Available: https://khn.org/news/death-by-a-thousand-clicks/. [Accessed 21 March 2023].

[20] D. F. Sittig et al., "Current Challenges in Health Information Technology-related Patient Safety," Health Informatics J., vol. 26, pp. 181-189, 2020.

[21] J. Grudin, J. Carroll, S. Ehrlich, M. Grisham, and H. Hersh, "Integrating human factors and software development," in CHI '88: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Washington D.C., 1988, pp. 157–159.

[22] A. Kushniruk, C. Nohr, and E. Borycki, "Human Factors for More Usable and Safer Health Information Technology: Where Are We Now and Where do We Go from Here?," Yearb Med Inform, pp. 120–125, 2016.

[23] E. M. Campbell, D. F. Sittig, J. S. Ash, K. P. Guappone, and R. H. Dykstra, "Types of Unintended Consequences Related to Computerized Provider Order Entry," J Am Med Inform Assoc., vol. 13, no. 5, pp. 547–556, 2006.

 ISBN: 978-1-68558-037-7 