



## OPEN ACCESS

## EDITED BY

Mark Sendak,  
Other, United States

## REVIEWED BY

Carly Bobak,  
Dartmouth College, United States  
Thankam Paul Thyvalikakath,  
Indiana University, United States

## \*CORRESPONDENCE

Azadeh Assadi  
az.assadi@mail.utoronto.ca

<sup>†</sup>These authors share senior authorship

## SPECIALTY SECTION

This article was submitted to Health Informatics, a section of the journal Frontiers in Digital Health

RECEIVED 29 April 2022

ACCEPTED 18 July 2022

PUBLISHED 04 August 2022

## CITATION

Assadi A, Laussen PC, Goodwin AJ, Goodfellow S, Dixon W, Greer RW, Jegatheeswaran A, Singh D, McCradden M, Gallant SN, Goldenberg A, Eytan D and Mazwi ML (2022) An integration engineering framework for machine learning in healthcare. *Front. Digit. Health* 4:932411. doi: 10.3389/fdgth.2022.932411

## COPYRIGHT

© 2022 Assadi, Laussen, Goodwin, Goodfellow, Dixon, Greer, Jegatheeswaran, Singh, McCradden, Gallant, Goldenberg, Eytan and Mazwi. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# An integration engineering framework for machine learning in healthcare

Azadeh Assadi<sup>1,2\*</sup>, Peter C. Laussen<sup>3,4</sup>, Andrew J. Goodwin<sup>1,5</sup>, Sebastian Goodfellow<sup>6</sup>, William Dixon<sup>1</sup>, Robert W. Greer<sup>1</sup>, Anusha Jegatheeswaran<sup>7</sup>, Devin Singh<sup>8,9</sup>, Melissa McCradden<sup>10,11,12</sup>, Sara N. Gallant<sup>1</sup>, Anna Goldenberg<sup>12,13,14,15</sup>, Danny Eytan<sup>1,16,17†</sup> and Mjaye L. Mazwi<sup>1,3,8†</sup>

<sup>1</sup>Department of Critical Care Medicine, Hospital for Sick Children, Toronto, ON, Canada, <sup>2</sup>Institute of Biomaterials and Biomedical Engineering, Department of Engineering and Applied Sciences, University of Toronto, Toronto, ON, Canada, <sup>3</sup>Institute of Medical Sciences, University of Toronto, Toronto, ON, Canada, <sup>4</sup>Executive Vice President for Health Affairs, Boston Children's Hospital, Boston, MA, United States, <sup>5</sup>School of Biomedical Engineering, University of Sydney, Sydney, NSW, Australia, <sup>6</sup>Department of Civil and Mineral Engineering, Faculty of Applied Science and Engineering, University of Toronto, Toronto, ON, Canada, <sup>7</sup>Department of Surgery, Division of Paediatric Cardiac Surgery, Hospital for Sick Children, Toronto, ON, Canada, <sup>8</sup>Translational Medicine, Peter Gilgan Centre for Research & Learning, Toronto, ON, Canada, <sup>9</sup>Department of Emergency Medicine, The Hospital for Sick Children, Toronto, ON, Canada, <sup>10</sup>Department of Bioethics, The Hospital for Sick Children, Toronto, ON, Canada, <sup>11</sup>Division of Clinical and Public Health, Dalla Lana School of Public Health, Toronto, ON, Canada, <sup>12</sup>Genetics & Genome Biology, Peter Gilgan Centre for Research & Learning, Toronto, ON, Canada, <sup>13</sup>Department of Computer Science, University of Toronto, Toronto, ON, Canada, <sup>14</sup>Vector institute for Artificial Intelligence, University of Toronto, Toronto, ON, Canada, <sup>15</sup>CIFAR, Toronto, ON, Canada, <sup>16</sup>Department of Medicine, Technion, Haifa, Israel, <sup>17</sup>Department of Pediatric Critical Care, Rambam Medical Center, Haifa, Israel

**Background and Objectives:** Machine Learning offers opportunities to improve patient outcomes, team performance, and reduce healthcare costs. Yet only a small fraction of all Machine Learning models for health care have been successfully integrated into the clinical space. There are no current guidelines for clinical model integration, leading to waste, unnecessary costs, patient harm, and decreases in efficiency when improperly implemented. Systems engineering is widely used in industry to achieve an integrated *system of systems* through an interprofessional collaborative approach to system design, development, and integration. We propose a framework based on systems engineering to guide the development and integration of Machine Learning models in healthcare.

**Methods:** Applied systems engineering, software engineering and health care Machine Learning software development practices were reviewed and critically appraised to establish an understanding of limitations and challenges within these domains. Principles of systems engineering were used to develop solutions to address the identified problems. The framework was then harmonized with the Machine Learning software development process to create a systems engineering-based Machine Learning software development approach in the healthcare domain.

**Results:** We present an integration framework for healthcare Artificial Intelligence that considers the entirety of this *system of systems*. Our proposed framework utilizes a combined software and integration engineering approach and consists of four phases: (1) Inception, (2) Preparation, (3) Development, and (4) Integration. During each phase, we

present specific elements for consideration in each of the three domains of integration: *The Human, The Technical System, and The Environment*. There are also elements that are considered in the interactions between these domains.

**Conclusion:** Clinical models are technical systems that need to be integrated into the existing *system of systems* in health care. A systems engineering approach to integration ensures appropriate elements are considered at each stage of model design to facilitate model integration. Our proposed framework is based on principles of systems engineering and can serve as a guide for model development, increasing the likelihood of successful Machine Learning translation and integration.

#### KEYWORDS

Integration engineering, artificial intelligence, machine learning, digital health, system of systems (SoS), human factors engineering (HFE), healthcare (MeSH)

## Glossary of key terms used in this manuscript

**Artificial Intelligence (AI):** A field that combines computer science and robust datasets to solve problems. These systems can be said to think like humans, act like humans, think rationally, or act rationally (1).

**Food and Drug Administration (FDA):** Regulatory body of the government responsible for maintaining health and safety of humans and animals within the United States through regulating food, drug, and technology (2).

**Human Factors Engineering (HFE):** A field of engineering focusing on the design of technology tailored to people as well as sociotechnical integration involving large, complex systems such as healthcare (3, 4).

**ISO/IEC/IEEE:** Global product, technological, procedural, and engineering standards set by globally recognized ISO/IEC/IEEE organizations (5, 6).

**Machine Learning (ML):** A branch of artificial intelligence and computer science that describes the ability of an algorithm to “learn” by finding patterns in large datasets (7).

**Software Development Lifecycles (SDLC):** The various software development frameworks that are used to structure software development (8).

**System of Systems (SOS):** A system that is composed of other systems and its “elements are managerially and/or operationally independent” (9).

## Introduction

Artificial Intelligence offers transformational opportunities in medicine, but this potential remains limited by a translation gap (10, 11). There are a variety of drivers that contribute to this gap. First, restrictions in data sharing limit training, validation and improvement of models (12). Second, lack of data and model transparency limit clinicians’ ability to interpret the model and evaluate it for relevance, accuracy and bias

impacting their trust in the model and thereby limiting utilization (12–14). Third, the absence of established model verification processes impose further challenges (12–14). Financial constraints, limited physician training in the field and rules and regulations that often lag technological advances further affect model integration (12–14). While the Food and Drug Administration has proposed guidelines to regulate some clinical models (15, 16), there are no current guidelines for clinical model integration. The term “model integration” is a more appropriate term than “implementation” as it recognizes that Artificial Intelligence models need to be compatible with the complex sociotechnical environments that characterize healthcare. Integration is defined as “an act or instance of combining into an integral whole” and refers to combining several implemented elements to form a fully realized system that enables interoperability between the various elements of the system (9). Improper integration of new systems may lead to additional costs, patient harm, damage to other systems, and decrease in efficiency (9). To address this translation gap, we present a systems engineering framework to guide the development of models with explicit consideration of elements that are crucial for successful model integration in healthcare.

## Foundations

### Systems engineering

Systems engineering is an interdisciplinary approach to system design to ensure its interactive elements are organized to achieve the purpose of the system (17, 18). The term dates to the early 1940’s and Bell Telephone Laboratories where it was used during World War II (19). The need for systems engineering came from the discovery that satisfactory components did not necessarily combine to produce a satisfactory system (20). This was particularly a problem for industries which produced complex systems at an early date, such as communications and aircraft industries (20). Systems

engineering forms the foundations of ISO/IEC/IEEE and is currently applied in a wide variety of industries from manufacturing to engineering and aerospace (18). A system is defined as “an aggregation of elements organized in some structure to accomplish system goals and objectives, is usually composed of **humans and machines** and has a definable structure and organization with external boundaries that separate it from elements outside the system (4).” A Machine Learning model should really be considered as a system composed of the model and data sources, its users, and its context. At higher degrees of abstraction, a system can be composed of other systems to create a system of systems (SOS), defined as a system whose “elements are managerially and/or operationally independent” (9). The healthcare system is a SOS (Figure 1) that results from the integration of the medical system, the regulatory, legal, and ethical systems, the financial system, the hospital system, the electronic health records, and many others. Therefore, for applied Machine Learning to become part of this existing SOS, it must be effectively integrated. Integration is the key to viability of any SOS (21) and achieving it requires effective collaboration

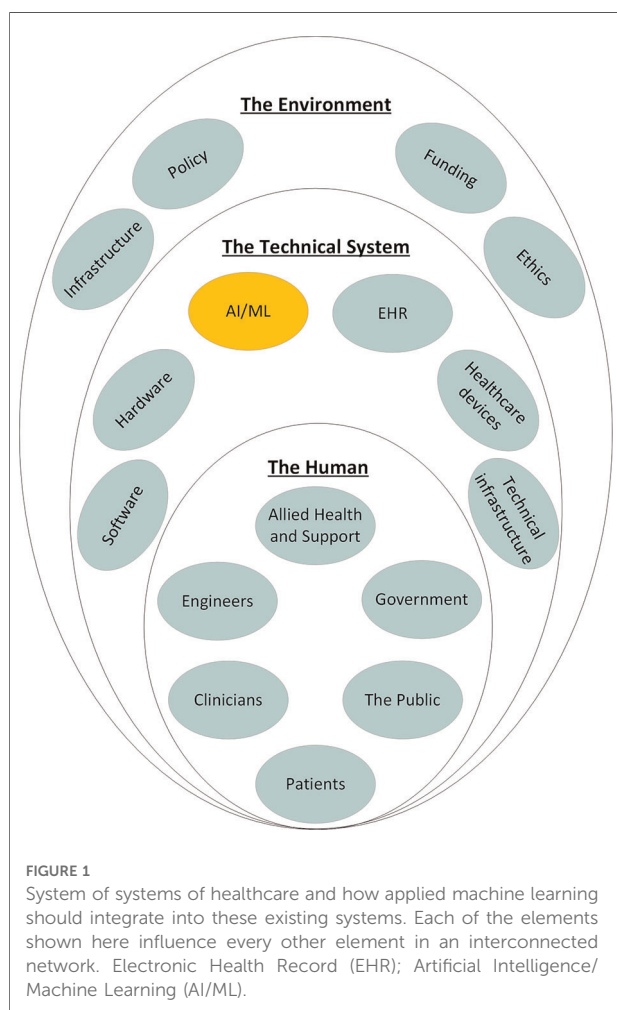
between these inter-operable systems (9). Integration, therefore, is a process that is analyzed, planned, designed, developed, executed, managed, and monitored throughout the system’s entire lifecycle and not just a distinct phase at the end (9).

Integration should consist of integrating the technical aspects of the system as well the human-system aspects. Technical integration of systems ensures that the various technical aspects of the system can work together to achieve the common objective (9). In the context of applied Machine Learning, this requires the interoperability of the models with the existing hardware and software infrastructure. To facilitate the technical integration of software, modular programming styles as well as Software Development Life Cycle (SDLC) frameworks have mechanisms that require developers to understand the existing systems and evaluate how the new system fits within this existing system as they progress through iterative development cycles. This is particularly important in dynamic industries like healthcare. The human-system integration aspect, also known as the sociotechnical integration, refers to the integration of a system within the social construct of the environment (9). Social demands as well as the societal or cultural values can play a major role in determining optimal performance of the entire system (22). Sociotechnical integration can be achieved through Human Factors Engineering (HFE), which is a field of engineering focusing on the design of technology tailored to people as well as sociotechnical integration involving large, complex systems such as healthcare (3, 4). The Systems Engineering Initiative for Patient Safety model provides a framework for integrating HFE in to healthcare to improve patient care quality and safety (23). Currently in its third iteration, this framework strongly advocates for a human centered design approach to engineering various aspects of the health care system such that the needs of patient and the people who care for these patients are put at the center of the design process (23).

Within systems engineering, the domains of integration are *The Technical System*, *The Human*, and *The Environment* and the interactions between them which have been previously described in literature (24) and summarized in Table 1. While this framework for integration has been used in the industry, it has not been applied to integrating Machine Learning in healthcare.

### Systems engineering in software development

Systems engineering principles are applied in software development. The SDLC defines the development process of software (8). It also relates to the architecture of the software and facilitates an understanding of the required resources for the software (8). The use of agile software development



**FIGURE 1** System of systems of healthcare and how applied machine learning should integrate into these existing systems. Each of the elements shown here influence every other element in an interconnected network. Electronic Health Record (EHR); Artificial Intelligence/ Machine Learning (AI/ML).

TABLE 1 Domains of integration and the interaction between them.

Domains of Integration	Definition
The Technical System	“An aggregation of elements organized in some structure to accomplish system goals and objectives, is usually composed of humans and machines and has a definable structure and organization with external boundaries that separate it from elements outside the system” (4)
Human	“An individual, a group of individuals, or organizations which have connections to the system in the form of owners, users, operators, managers, service providers, supplies, producers, or other stakeholders, who directly or indirectly have an interest in the system.” (9)
Environment	“All the relevant parameters that can influence or be influenced by the system in any lifecycle phase.” (9)
System-Environment interaction	A physical interaction occurs through technical interfaces while a non-physical interaction can occur through laws, regulations, policy, market demands and political interests, which may influence or be influenced by the system (9).
Human-System interaction	The physical, logical, or emotional relationship between the human and the system that can be influenced by or influence the system. HFE largely aims to optimize this interaction (9, 25).
Human-Environment interaction	Relationship between the human and the internal and external workplace or system environment. Some examples include organizational attributes that may affect decision-making processes of humans, circumstances that may cause deviation from standard operating procedures, impact of noise, temperature, illness, fatigue, interpersonal relationships, etc. can also influence the system or be influenced by it (9, 25). HFE can also be used here to optimize some of these challenges.

HFE, Human Factors Engineering.

techniques has facilitated software implementation but is seldom used by healthcare Machine Learning model developers. This approach also fails to recognize some of the unique implications of applied Machine Learning in practice. The purpose of SDLC has evolved over its 60-year history from ensuring an understanding of what needs to be done to focusing on structured development methods, to focusing on product delivery (26). To achieve this, there needs to be a balance between the structured and agile SDLC frameworks (26). The evolution of these different life cycles has been in response to the increasing complexity of software, the systems for which new software are being designed, advancements in hardware, and the widespread use of software in society. Despite successful use of software globally, a strong emphasis on time to market has led to the incomplete application of many well-known SDLC recommendations, particularly those of requirement gathering, planning, specifications, architecture, design, and documentation (27). There are also inherent limitations to each of the SDLC models that can further contribute to poor software design (Table 2). In addition to cost, technical concerns, need for workflow alterations, privacy concerns, perceived lack of usefulness, productivity loss, and usability issues have contributed to the very slow uptake of healthcare software such as electronic health record systems in United States (30).

## Challenges of machine learning and existing software development life cycle

There are multiple frameworks proposed for Machine Learning development that essentially focus on *context understanding*, *data curation*, *data modeling*, and *production and monitoring* (31–34). Yet, despite relatively fast and cheap

development and deployment of Machine Learning models, they have been difficult and expensive to maintain and integrate (35). This is in part due to existing challenges with medical software and in part due to some unique Machine Learning issues (35). Table 3 illustrates some of the Machine Learning challenges that impact its development.

## Challenges with machine learning model integration into healthcare

Developing Machine Learning models for healthcare imposes unique challenges that can impact successful clinical integration. Some challenges relate to the social complexity of medicine and others to the safety critical nature of medical systems. Table 4 summarizes some of the described challenges and gaps in clinical Artificial Intelligence models and the health care environment that limit their use. For example, health care data can be very noisy and as such, often subject to data preprocessing. This preprocessing may result in training and testing data that may not be representative of the “real world” data that the model will experience in practice (Table 3, M14; Table 4, C7). Variations in institutional Electronic Health Record and the information in the structure of this type of data also challenge model performance across institutions (45).

## Methods

A modified narrative review method was used to understand (a) the challenges and gaps in integrating Machine Learning models in health care, (b) challenges associated with Machine learning models and current SDLC, and (c) principles of

TABLE 2 Overview of current software development life cycle models and their limitations (8, 28, 29).

SDLC	Overview	Limitations
Classical Waterfall model	<ul style="list-style-type: none"> <li>- Series of processes in succession without gap</li> <li>- Foresees defect or fault</li> <li>- Requires proper planning and well-articulated documentation</li> <li>- “Characterize before the design”</li> <li>- Used in safety critical systems where phases and processes are inter-dependent and there is a high need for assurance with no tolerance for mistakes</li> </ul>	<ul style="list-style-type: none"> <li>- Not flexible</li> <li>- Prototypes made late in the overall process</li> <li>- Product delivery often delayed</li> <li>- High risk and uncertainty</li> <li>- Not suitable for complex and object-oriented projects</li> <li>- Not suitable for long and ongoing projects</li> <li>- Not suitable for existing systems</li> </ul>
Iterative Waterfall model	<ul style="list-style-type: none"> <li>- Use iterations to prototype and refine the project’s requirements before proceeding with the waterfall model for the rest of the development process</li> </ul>	<ul style="list-style-type: none"> <li>- Iteration is possible but predisposed to errors and costly</li> <li>- Not suitable for long term projects</li> <li>- Difficult to gather requirements</li> <li>- Changes in previous stages can cause big issues in subsequent stages</li> </ul>
Prototyping model	<ul style="list-style-type: none"> <li>- Leverage the use of prototypes to clarify and refine requirements</li> <li>- May use prototype to iteratively build the finished project or simply use it as a demonstration of what is being proposed as a solution</li> </ul>	<ul style="list-style-type: none"> <li>- Requires system modifications after implementation</li> <li>- Can increase complexity of the system</li> <li>- Leads to incomplete applications</li> </ul>
Evolutionary model	<ul style="list-style-type: none"> <li>- Requirements change over time and the initial design evolves with user interaction and input as well as with new requirements</li> </ul>	<ul style="list-style-type: none"> <li>- Planning and design phase are incomplete</li> <li>- Not suitable for incremental building</li> <li>- Costly</li> </ul>
Spiral model	<ul style="list-style-type: none"> <li>- Combination of top-down and bottom-up constructs</li> <li>- Can be used with other models</li> <li>- Breaks a project into smaller segments so simplify development and evaluation</li> <li>- For systems when cost and risk assessment are key</li> <li>- Also, when users are uncertain about their needs</li> </ul>	<ul style="list-style-type: none"> <li>- Costly</li> <li>- Requires high expertise for risk analysis</li> <li>- Risk analysis central to project success</li> <li>- Not suitable for small projects</li> </ul>
V-model	<ul style="list-style-type: none"> <li>- Focus on validation and verification—the product from each phase is checked and approved before moving on to the next phase</li> </ul>	<ul style="list-style-type: none"> <li>- Very rigid</li> <li>- Prototypes are available late in the development phase</li> <li>- Changes require lots of documentation</li> </ul>
RAD model	<ul style="list-style-type: none"> <li>- Rapid, iterative design of small parts of the project to put into test and ensure project on track and meeting requirements before pursuing the next iteration</li> <li>- Agile software development falls in this category</li> </ul>	<ul style="list-style-type: none"> <li>- Depends on strong member performance to identify requirements</li> <li>- Only suitable for modular systems</li> <li>- Requires very skilled developers with good modeling skills</li> <li>- Costly</li> </ul>

SCLD, Software Development Life Cycle; RAD, Rapid Application Development.

integration engineering from a systems engineering and human factors engineering perspective. A narrative review method is the searching of the literature with a specific goal in mind where manuscripts are hand-selected for inclusion based on the research questions (46). The ACM-DL, PubMed, and IEEE Xplore databases were used for this narrative review with the following search terms: (“All Metadata”:“machine learning” OR “All Metadata”:“artificial intelligence” OR “All Metadata”:“algorithm” OR “All Metadata”:“model”), (“All Metadata”:“healthcare” OR “All Metadata”:“medicine” OR “All Metadata”:“clinical” OR “All Metadata”:“health care” OR “All Metadata”:“health”), (“All Metadata”:“framework”), (“All Metadata”:“development”), and (“All Metadata”:“software

development lifecycle”). The search was restricted to publications in the last 5 years and original peer-reviewed research and reviews. Duplicate results were removed based on manuscript title, and relevant manuscripts were selected based on abstracts. The selected manuscripts were then reviewed, summarized, and synthesized to outline (a) challenges and gaps in integrating Machine Learning Models in health care, (b) challenges associated with Machine learning models and current SDLC, and (c) principles of integration engineering from a systems engineering and human factors engineering perspective.

To address some of the challenges associated with Machine Learning models and current SDLC as well as Machine

TABLE 3 Summary of challenges associated with machine learning life cycle (35).

Aspects of software Development	Features and challenges with Machine Learning development
Software Requirements	<ul style="list-style-type: none"> <li>- Uncertain requirements (conceptual description of the goal after applying Machine Learning systems; different data and different application context would lead to different requirements) [M1]</li> <li>- Quantitative measures such as accuracy define requirements with little regard to functional requirements (the exact desired quantitative measures (e.g., accuracy) are not always known) [M2]</li> <li>- Requirement validation requires a larger number of preliminary experiments, ideally with real data [M3]</li> <li>- Requirement must consider the predictable degradation in performance of Machine Learning systems (must be degradation-sensitive and adapt to degradation through ongoing training or re-training) [M4]</li> </ul>
Software Design	<ul style="list-style-type: none"> <li>- Insufficient emphasis on the coupling of components (e.g., quality of data processing and performance of Machine Learning models) [M5]</li> <li>- Flexible detailed design with need for multiple, iterative experimentation to develop an effective model [M6]</li> </ul>
Software Construction and Tools	<ul style="list-style-type: none"> <li>- Bulk of coding is focused on developing an effective Machine Learning model [M7]</li> <li>- Debugging focused on improving model performance (need real data and often delayed until last stages) [M8]</li> <li>- Debugging can take a very long time based on data size and complexity of a model [M9]</li> <li>- Bugs can be hidden in the data [M10]</li> </ul>
Software Testing and Quality	<ul style="list-style-type: none"> <li>- Hard to reproduce test results because of sources of randomness [M11]</li> <li>- Testing output are often a range or probability based rather than a single value [M12]</li> <li>- Quality of testing is highly dependent on the quality of the test case and testing dataset [M13]</li> <li>- Good testing results cannot guarantee performance in production or generalizability (highly dependent on similarities of training/testing datasets and the real-world data) [M14]</li> </ul>
Software Maintenance and Configuration Management	<ul style="list-style-type: none"> <li>- Expect performance degradation [M15]</li> <li>- Require configuration management to keep track of varying models and associated tradeoffs, algorithm choice, architecture, data, hyperparameters, etc. [M16]</li> </ul>
Software engineering Process and Management	<ul style="list-style-type: none"> <li>- Overestimation of what Machine Learning can do leading to mismatch of expectation and reality [M17]</li> <li>- Limited incorporation of domain expertise into the engineering and management process [M18]</li> <li>- Sustained performance requires ongoing monitoring and planned evaluation to determine timing to retrain and to rectify mistakes and unexpected consequences [M19]</li> <li>- No standard guidance for the management of Machine Learning development [M20]</li> </ul>

ML, Machine Learning.

Learning models for the health care environment, we adapted existing software development guidelines as well as principles of systems engineering to develop our framework described below. The framework was iteratively designed through synthesis of the literature with expert input from the research team in domains of human factors engineering, Machine Learning, medicine, and software development. Consensus was achieved on the final iteration of the framework which was organized in keeping with steps in Machine Learning model development. In the supplementary material, we apply this framework to the design of an arrhythmia detection model intended for clinical integration.

## Results

We present a generalizable framework (Figure 2) that identifies four phases in the development of Artificial Intelligence models in healthcare: (1) Inception, (2)

Preparation, (3) Development, and (4) Integration. Each phase incorporates considerations from the key domains of integration and systems engineering as well as the interaction between them for an integrated SOS as we show below. Outcomes from each phase while informing the phase that follows, also provides feedback to previous phases, particularly when there are new findings in a phase that were not previously considered. The challenges outlined in Table 3 that are addressed with our proposed model are indicated in [M#] format while challenges from Table 4 are indicated in [C#] as the features and steps are described. We have applied this framework to an arrhythmia detection model and have implemented this as a best practice at the Hospital for Sick Children. This practical application is demonstrated in the supplementary material. It is important to note, that the successful integration of a system into an existing system also requires ongoing maintenance and refinement which includes detection of performance degradation, changing workflows and policies, changing hardware and data acquisition, and

TABLE 4 Challenges with machine learning models in healthcare.

Aspects of Machine Learning-models and the healthcare environment	Gaps or Challenges
Context	<ul style="list-style-type: none"> <li>- Need to thoroughly understand the clinical data being used for model development (13, 36) [C1]</li> <li>- Need models with impactful clinical utility (13) [C2]</li> <li>- Need models that fit within the environment they are intended for (13, 37–41) [C3]</li> </ul>
Data	<ul style="list-style-type: none"> <li>- Need access and availability to well labeled, high quality, large datasets (13, 14, 39) [C4]</li> <li>- Need consistency in data collection techniques (13) [C5]</li> <li>- Need to acknowledge and minimize inaccurate or incomplete data (13, 41) [C6]</li> <li>- Need to ensure that model training/test data is representative of what the model will experience during operation; consider pre-processing of data and its effect [C7]</li> <li>- Need to identify, remove, and account for biased data (13, 14, 40, 41) [C8]</li> <li>- Need to account for data shifts and their effect on model performance [C9]</li> </ul>
Model validation and performance	<ul style="list-style-type: none"> <li>- Need to conduct and develop clinical validation studies (11, 13, 14, 37) [C10]</li> <li>- Need to conduct clinical impact/outcome studies as Machine Learning metrics (accuracy, precision, etc.) often do not map directly to clinical performance indicators (14, 37) [C11]</li> <li>- Need model transparency (11, 39, 41) [C12]</li> </ul>
Ethics and Regulation	<ul style="list-style-type: none"> <li>- Need regulation and safe use guidelines (14, 39, 42) [C13]</li> <li>- Need privacy and cybersecurity regulations (39–41, 43) [C14]</li> <li>- Need to screen for algorithmic biases (11) [C15]</li> </ul>
Financial issues	<ul style="list-style-type: none"> <li>- Need adequate resources (hardware, expertise, software, etc. all in high demand, limited, and expensive) to develop and integrate models (39) [C16]</li> </ul>
Knowledge gap	<ul style="list-style-type: none"> <li>- Need users to have sufficient knowledge to interpret model output or compare different models (11, 39, 41, 44) [C17]</li> </ul>

ML, Machine Learning.

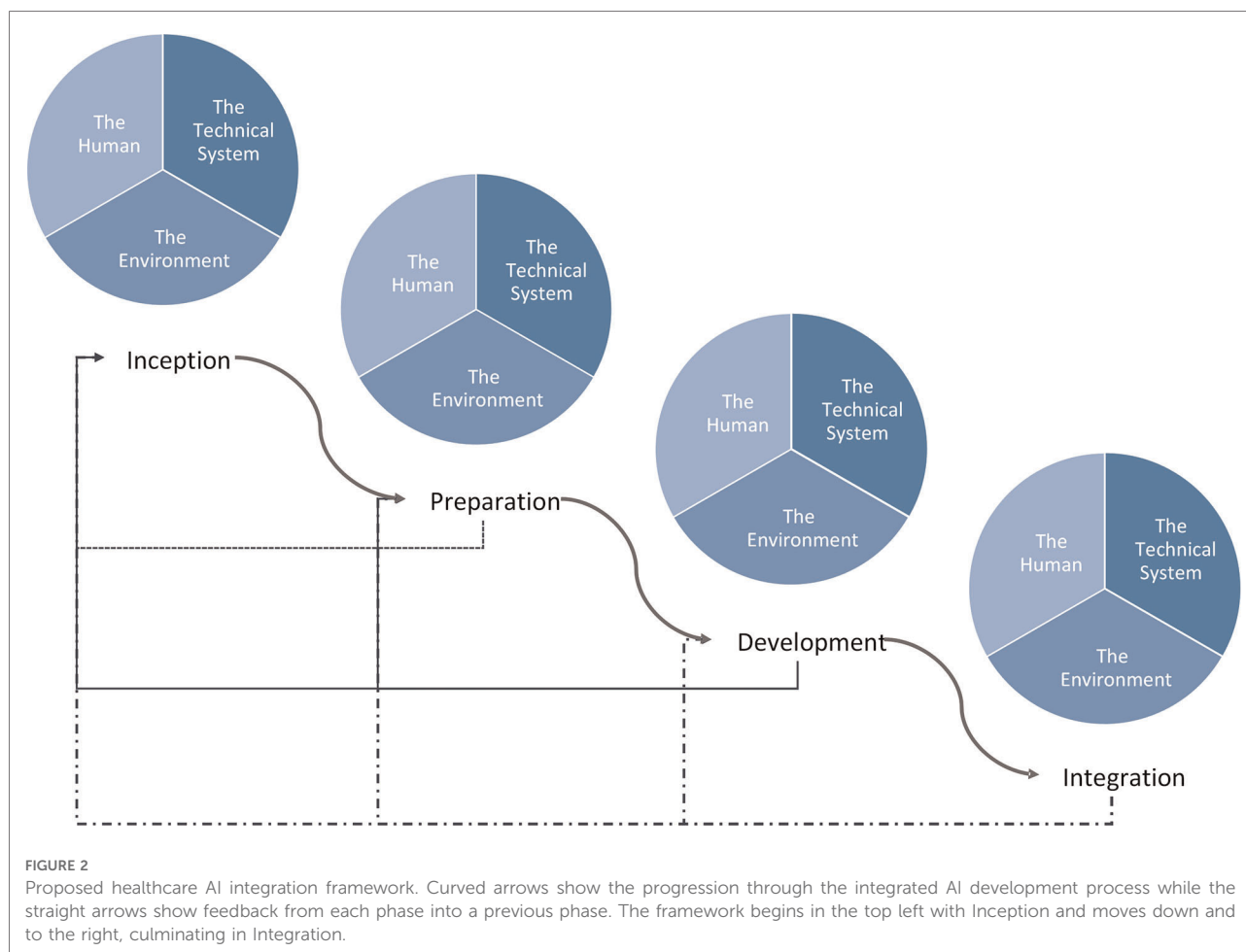
changing knowledge and familiarity with Artificial Intelligence [M15, M19, M20]. This maintenance phase is not included in our integration framework as this follows successful initial model integration which is our focus.

## Inception

Inception refers to the very first phase of model development during which an appropriate use case and modelling approach are identified. During this phase, specific considerations for the *Technical System* include clear problem definition and clinical applications [M1, C1 & C2], strategies, and techniques to address the problem [M1], the kinds of data the model would need and whether the data is available or can be made available [M5, M13 & M14], sourcing the data, and whether the data considered for training is similar to what would be used in the intended environment [M5, M13, M14, C1-4, C7]. Close collaboration between clinicians (the domain experts and future users), model developers, and data scientists is essential. The optimal way of achieving this collaboration is through the expertise of project managers, product owners, and business analysts who specialize in gathering requirements, documenting specifications, uncovering pain points, defining business key

performance indicators [M1-4, M16, M18, C1-3, C10-11, C14, C16, C17]. Given the need for a wide variety of expertise, close collaboration of a multidisciplinary team is central to the successful design and development of Artificial Intelligence models for healthcare. Our proposed inception phase incorporates integration considerations into the *context understanding* phase of current Machine Learning model development lifecycles.

*The Human* considerations during this phase include identification of all stakeholders (clinicians, data scientists, and modeling experts) and clear problem definition [M17, M18, C1 & C2]. As part of defining the problem, it is important to identify existing human challenges with the problem (e.g., who is affected by the problem, how and why), any previous attempts at addressing the problem (e.g., what has been tried, what has worked, what failed and why), and ensuring a Machine Learning solution is suitable to the problem. Some potential strategies to achieve these objectives at this phase include stakeholder engagement, informal focus groups, and immersion in the problem space. Early involvement of stakeholders provides them with a greater understanding of the limitations of Machine Learning as well as investment in model evaluation, knowledge translation, integration, and monitoring (4, 25). Similarly, Machine Learning model developers and data



scientists can also gain insight into the nature of the medical problem they are solving, as well as the unique features and properties of the data they would analyze and the environment they would be designing for.

Some *Environmental* considerations during this phase include an understanding of environmental constraints, the kinds of necessary hardware for the model training and operation, and data storage. This assessment of the environment can allow for more accurate estimates of cost as it relates to the clinical integration of the model which can facilitate estimations on feasibility of integration as well as future evaluation of balancing measures and project costs [C16]. **Supplementary Table S1** illustrates the practical application of this phase of our proposed framework as it relates to the arrhythmia detection model.

## Preparation

The *Technical System* related considerations during the preparation phase ensure relevant data are consistently,

accurately, and reliably acquired and labelled for model training and evaluation [C4-7]. This is particularly important in clinical data as high-quality labels require the expertise of clinicians who are often restricted by availability. In addition, disagreements between multiple clinicians labeling the data can also introduce noise to the data and impact model training [C4-5, C7]. Preliminary analysis is also done on the “real” data to ensure it is suitable for the model as imagined [M3, M5, C1, C7] as well as ensuring any systematic bias or inaccuracies are identified and addressed [C8, C15]. This phase combines the current model development lifecycle phases of *data curation*, *data modeling* while also incorporating integration considerations.

*Human* considerations include the completion of a formal needs assessment, as well as cognitive and workflow analysis to identify specific needs and inefficiencies [M1, C2, C3]. User requirements are further defined and clarified in this phase through engagement with representatives from all stakeholders [M1, M18]. This can be achieved through cognitive task analysis, task analysis, workflow analysis, focus groups, interviews, and simulations. This establishes expectations which are calibrated during model development. Knowledge gaps



among model users should be formally studied and identified to guide the development of a knowledge translation plan in the Development phase [C17]. This includes knowledge gaps in the clinical field for which a model is being designed for.

*Environmental* considerations in this phase include an evaluation of existing privacy and data security measures, as well as ethical and policy regulations that may need to be further developed to facilitate model integration and clinical utility. [Supplementary Table S2](#) illustrates the practical application of this phase of our proposed framework as it relates to the arrhythmia detection model.

## Development

During this phase, the *Technical System* (the model and its associated user interface) is developed through iterative design, testing, and statistical evaluations. This process should mimic that of agile software design (RAD model) with rapid modeling and testing of the model output evaluated against clinical gold standards and with real production datasets [M6, M7, M14, C7 & C11]. Other SDLC models can be more expensive, slower to iterate and develop, and present fewer opportunities for user and stakeholder engagement throughout the development cycles. This rapid and iterative development and testing would allow early and real-time feedback on the performance of the model [M8-10, M13, & M17] and generate performance matrices that can be used in testing with users to determine acceptable ranges of performance based on the clinical context of the application [M2, & M12]. The results of these assessments and the features of the model at each stage should be recorded to allow future auditing as well as the establishment of a track record and rigor that would be important in establishing clinician trust [M16]. Model output should also be reproducible through thoughtful selection of training and testing data instead of random sampling [M11]. These objectives can be realized through iterative retrospective studies as well as prospective silent trials that can be coupled with simulation testing to evaluate some of the *system-human* interaction considerations. If explainable Artificial Intelligence is required to achieve transparency of a model and facilitate its clinical integration, the explanations should also be developed during this time and evaluated together with the technical system itself for both accuracy and relevance, as well as usability and impact on decision making as described below [C12, C16]. Other means of achieving transparency include ensuring data properties (including pre-processing techniques), algorithmic properties, validation testing results, and other properties are well documented and disclosed to users as needed [C12]. These would be most relevant at the initial adoption phase of a model until its performance is experientially understood by users. The *production* phase of

current model development lifecycles correlates to this new proposed phase.

The technical system is also refined based on considerations of the *Human* which involve a variety of formative and summative assessments that ensure user centered design and a robust knowledge translation strategy. Simulations and other human factors assessments should be used during this phase to evaluate the model's fit within the cognitive schema of users and their existing or proposed workflow. These investigations allow for further improvements to the model. Based on the stage of the model development, simulations of varying fidelity, from low fidelity tabletop activities to evaluate workflow and the model's user interface, to high fidelity in person simulations done in near-live environments with potentially real or realistic patient data and scenario can be conducted. Another opportune moment to understand relevant interactions is during silent trials where the model is run on real data, but its output is not made visible to the clinicians providing direct patient care. At the same time, the model can be made available to off-service clinician representatives from the previously defined user groups, and their interactions with the tool as well as its effect on their decision making, workflow, efficiency, accuracy, team dynamics, and much more can be studied and evaluated [C3]. The results of such a study can be fed back to the system developers to further optimize the system as well as used by policy makers and HFEs to refine the environment, training materials, education sessions, policies and much more. Depending on the complexity of the system and the fidelity of the simulation, these studies should be repeated until there is consistency and satisfaction in the performance of the SOS. It's important to note that through these simulations, broader user engagement can be achieved which would serve as a medium through which the stakeholders are made familiar with a tool that is under development as well as the ways in which it is being proposed for use clinically [M17-19, & C16].

The *Environmental* considerations during this phase ensure the development and security of software and hardware that facilitate optimal model operation within its intended environment. Infrastructure is also evaluated and optimized for various failure modes, inefficiencies, and instabilities. Necessary policies, regulations, and rules of engagement with the model, as well as ethical and privacy guidelines would also be developed and evaluated during this phase of model development [C13, C14]. [Supplementary Table S3](#) illustrates the practical application of this phase of our proposed framework as it relates to the arrhythmia detection model.

## Integration

The *Technical System* at this phase should be performing optimally for its intended problem, stakeholders, and

healthcare context. During this phase, the model is launched for a prospective clinical evaluation [C10] and further refined based on live performance and user feedback [M19]. These prospective clinical evaluations should be subject to the peer review process to facilitate model adoption [C10]. Prospective studies should also be conducted to evaluate any long-term effects of model adoption (e.g., performance degradation secondary to practice changes) [C10] as well as any indications to suspect algorithmic biases [C14]. Finally, system monitoring should be ensued. This includes monitoring system metrics (e.g., server load, throughput, latency, etc.), input metrics (e.g., number of missing values, failed event detections, minimums, maximums, means, standard deviations, central frequencies, etc.), and output metrics (e.g., null predictions, model confusion, rate of change, etc.). It should also be evaluated for possible effects of data shift secondary to practice driven changes in data which may be due to the use of the model itself [M15, C9].

The *Human* considerations involve the launch of the developed knowledge translation plan prior to model deployment with ongoing just-in-time training when the model is clinically deployed. User engagement in the previous stages to develop knowledge translation strategies as well as evaluate model performance and usability would have allowed opportunities to calibrate user expectations to model capabilities and performance. Nevertheless, a formal knowledge translation process before the launch of the technical system and during the integration phase will further calibrate these expectations and establish a functional understanding of the technical system's operation and place within the existing workflow [M1, M17, C17]. Ongoing cost, workflow, and cognitive assessments are also leveraged as a feedback mechanism to further refine the overall system [C2].

The *Environmental* considerations during this phase, in addition to rolling out the developed and tested infrastructure as well as policies, procedures, privacy, and ethical considerations should also include an ongoing evaluation and refinement of these in practice. [Supplementary Table S4](#) illustrates the practical application of this phase of our proposed framework as it relates to the arrhythmia detection model.

Upon the completion of this last phase, the Artificial Intelligence system is expected to be fully integrated into its intended healthcare environment and ready for clinical use.

## Discussion

The healthcare environment is a complex SOS with multiple integrated systems from a wide variety of domains. Recognizing this complexity, our framework takes a systems engineering approach to Machine Learning model design for integration.

The interdisciplinary approach promoted by systems engineering ensures that the interactive components of a

system are organized to achieve the purpose of the system (9, 10). For Machine Learning system design in the healthcare space, utilizing clinical, legal, ethical, and human factors expertise is as important as ensuring adequate Machine Learning, data science, and infrastructure expertise in the design and development process. Therefore, these aspects of a healthcare Machine Learning system are incorporated from inception and the involvement of experts from these different disciplines are expected throughout the development lifecycle of such systems in our framework. Involving project managers and business analysts, for example, can further strengthen the collaboration among different domain experts and facilitate the development of a well-designed product. End-user engagement is also emphasized in our framework as early and close engagement with the development of Machine Learning systems for healthcare will lead to improvements in system performance and fit. Greater transparency from this early and collaborative engagement, as well as user driven development of model explanations contribute to end-users' trust and adoption of Machine Learning.

As the number of integrated models for the healthcare domain increases and as users become increasingly familiar with Machine Learning models in healthcare, the need to integrate models designed in other clinical environments and institutions will increase. Our proposed framework could be used to evaluate these existing models for their fit into the new SOS. Issues with performance due to differences in data properties, fit in the workflow, usability, policies, environmental space, hardware requirements and others can be identified through the steps in this framework. Solutions to these issues can then be developed and tested until deemed adequate before the model is integrated into the new SOS.

Finally, the transformational opportunities that Machine Learning, and more broadly Artificial Intelligence, offer the field of medicine and healthcare, range from improving quality and efficiency in healthcare, improving accessibility, personalizing medicine, to advancing the field of medicine and healthcare. These models are fundamentally different to the current technology used in healthcare and they can not only move research in medicine from a hypothesis driven model to one that is data driven, but also modify clinical decision-making to be more data driven as well (47). The rigorous development and integration of these models is therefore crucial in maximizing their beneficial impact on healthcare.

## Limitations

The framework that is proposed here aims at guiding the development of models for healthcare from their inception to their integration into the intended clinical space. For systems to be successful, they should also have a maintenance plan which would allow for ongoing modifications to optimize the

system within its broader SOS in response to rare scenarios that may not have been encountered during the development phase as well as changes in the SOS that occur over time. For Machine Learning models, this is particularly relevant as the introduction of new technologies and changing practice patterns can have significant implications for the data based on which these models operate, potentially leading to a deterioration in model performance. The degradations in model performance should be continuously monitored and acted upon to ensure ongoing acceptable performance and reliability in their clinical application. To ensure due attention to integration without overlooking the important features in integration and maintenance, the detailed description of a maintenance phase for health care models was left out of this manuscript. We intend on exploring and developing a comprehensive maintenance phase in our future work. This will include guidelines on monitoring and maintaining longitudinal key performance indicators and model performance, as well as establishing thresholds for model retraining, workflow modifications, user re-training, and environmental modification (e.g., new policies, technology, and laws).

## Conclusion

Artificial Intelligence models in healthcare are technical systems that need to be integrated into an existing system of systems that also includes the human and the environment. An integration engineering approach allows the creation of a pragmatic framework that we believe will both address the translation gap and inform and support regulatory approaches to Artificial Intelligence models in healthcare.

## Data availability statement

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

## References

1. IBM Cloud Education. Artificial intelligence (2020). Available at <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence> (cited June 24, 2022).
2. FDA. FDA: What we do (2018). Available at <https://www.fda.gov/about-fda/what-we-do> (cited June 24, 2022).
3. Vincente K. *The human factor*. New York, USA: Routledge (2006).
4. Salvendy G. *Handbook of human factors and ergonomics*. 4th ed. New Jersey: John Wiley & Sons, Inc. (2012). 1–1736.
5. IEEE SA. About Us. Available at <https://standards.ieee.org/about/> (cited June 24, 2022).
6. ISO. Standards. Available at <https://www.iso.org/standards.html> (cited June 24, 2022).
7. Rowe M. An Introduction to machine learning for clinicians. *Acad Med.* (2019) 94(10):1433–6. doi: 10.1097/ACM.0000000000002792
8. Akinsola JET, Ogunbanwo AS, Okesola OJ, Odun-Ayo IJ, Ayegbusi FD, Adebisi AA. Comparative Analysis of Software Development Life Cycle Models (SDLC) (2020). 310–22. Available at [http://link.springer.com/10.1007/978-3-030-51965-0\\_27](http://link.springer.com/10.1007/978-3-030-51965-0_27)
9. Rajabalinejad M, van Dongen L, Ramtahalsing M. Systems integration theory and fundamentals. *Saf Reliab.* (2020) 39(1):83–113. doi: 10.1080/09617353.2020.1712918

## Author contributions

AA, PCL, DE, and MM closely collaborated to develop the presented framework. AG, SGoodfellow, WD, and RG in collaboration with DE developed the arrhythmia detection model and provided technical insight into the features of the framework pertaining to model development and evaluation. AG also provided additional overarching technical insight. AJ and DS in collaboration with AA, PCL, DE, and MM provided clinical insight into development of the framework while MC provided the ethical insight and considerations. SG provided insights pertaining to cognitive load. All authors contributed to the article and approved the submitted version.

## Conflict of interest

MMcCradden is the John and Melinda Thompson Chair of Artificial Intelligence in Medicine and acknowledges funding from the SickKids Foundation relating to this role. All other authors have no conflicts of interest pertaining to this work to disclose.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fdgth.2022.932411/full#supplementary-material>.

10. Keane PA, Topol EJ. With an eye to AI and autonomous diagnosis. *npj Digit Med.* (2018) 1(1):10–2. doi: 10.1038/s41746-018-0048-y
11. Kelly CJ, Karthikesalingam A, Suleyman M, Corrado G, King D. Key challenges for delivering clinical impact with artificial intelligence. *BMC Med.* (2019) 17(1):1–9. doi: 10.1186/s12916-019-1426-2
12. He J, Baxter SL, Xu J, Xu J, Zhou X, Zhang K. The practical implementation of artificial intelligence technologies in medicine. *Nat Med [Internet].* (2019) 25(1):30–6. doi: 10.1038/s41591-018-0307-0
13. Wiens J, Saria S, Sendak M, Ghassemi M, Liu VX, Doshi-Velez F, et al. Do no harm: a roadmap for responsible machine learning for health care. *Nat Med.* (2019) 25(9):1337–40. doi: 10.1038/s41591-019-0548-6
14. Rajkomar A, Dean J, Kohane I. Machine learning in medicine. *N Engl J Med.* (2019) 380(14):1347–58. doi: 10.1056/NEJMr1814259
15. Food US, Administration D. Software as a medical device (SAMD): clinical evaluation guidance for industry and food and drug administration staff. *FDA Guid.* (2017):1–32. Available at <https://www.fda.gov/media/100714/download>
16. Medical Device Coordination Group. Guidance on Qualification and Classification of Software in Regulation (EU) 2017/745—MDR and Regulation (EU) 2017/746—IVDR (2019):28. Available at <https://ec.europa.eu/docsroom/documents/37581>
17. ISO/IEC/IEEE. Systems and Software Engineering—System life cycle processes. ISO/IEC/IEEE 15288 (2015):1.
18. Friedenthal S, Moore A, Steiner R. Systems engineering overview. In: *A Pract Guid to SysML [Internet].* Third. Elsevier (2015) p. 3–14. <https://linkinghub.elsevier.com/retrieve/pii/B9780128002025000011>
19. Buede DM. The engineering design of systems [Internet]. In: M Dennis, J Wiley, editors. *The engineering design of systems: models and methods.* Hoboken, NJ, USA: John Wiley & Sons, Inc. (2009). p. 1–48. <http://doi.wiley.com/10.1002/9780470413791>
20. Schlager KJ. Systemas engineering-key to modern development. *IEEE Trans Eng Manag.* (1956) EM-3(3):64–6. doi: 10.1109/IRET-EM.1956.5007383
21. Jamshidi M. System of systems engineering—new challenges for the 21st century. *IEEE Aerosp Electron Syst Mag.* (2008) 23(5):4–19. Available at <http://ieeexplore.ieee.org/document/4523909/> doi: 10.1109/MAES.2008.4523909
22. Woo DM, Vicente KJ. Sociotechnical systems, risk management, and public health: Comparing the North Battleford and Walkerton outbreaks. *Reliab Eng Syst Saf.* (2003) 80(3):253–69. Available at <https://linkinghub.elsevier.com/retrieve/pii/S0951832003000528> doi: 10.1016/S0951-8320(03)00052-8
23. Carayon P, Wooldridge A, Hoonakker P, Hundt AS, Kelly MM. SEIPS 3.0: human-centered design of the patient journey for patient safety. *Appl Ergon.* (2020) 84:103033. Available at <https://linkinghub.elsevier.com/retrieve/pii/S000368701930239X> doi: 10.1016/j.apergo.2019.103033
24. Rajabalinejad M. Incorporation of safety into design by safety cube. *Int J Ind Manuf Eng.* (2018) 12(3):476.
25. American Association for Medical Instrumentation. AAMI HE75: 2009(R 2018) | Human Factors Engineering—Design of Medical Devices (2018). Available at [https://infostore.saiglobal.com/en-au/Standards/AAMI-HE75-2009-R-2018-517\\_SAIG\\_AAMI\\_AAMI\\_2628453/](https://infostore.saiglobal.com/en-au/Standards/AAMI-HE75-2009-R-2018-517_SAIG_AAMI_AAMI_2628453/)
26. Kneuper R. Sixty years of software development life cycle models. *IEEE Ann Hist Comput.* (2017) 39(3):41–54. doi: 10.1109/MAHC.2017.3481346
27. Shafiq S, Mashkour A, Mayr-Dorn C, Egyed A. A literature review of using machine learning in software development life cycle stages. *IEEE Access.* (2021) 9:140896–920. doi: 10.1109/ACCESS.2021.3119746
28. Kumar M, Rashid E. An efficient software development life cycle model for developing software project. *Int J Educ Manag Eng.* (2018) 8(6):59–68. doi: 10.5815/ijeme.2018.06.06
29. Stephens R. *Beginning software engineering.* Indianapolis: John Wiley & Sons, Inc. (2015).
30. Kruse CS, Kristof C, Jones B, Mitchell E, Martinez A. Barriers to electronic health record adoption: a systematic literature review. *J Med Syst.* (2016) 40(12):252–8. doi: 10.1007/s10916-016-0628-9
31. Mishra A. Amazon Machine learning. *Mach Learn AWS Cloud.* (2019):317–51. doi: 10.1002/9781119556749.ch15. Available at <https://docs.aws.amazon.com/machine-learning/latest/dg/the-machine-learning-process.html>
32. Microsoft Azure. What is the Team Data Science Process? (2022). Available at <https://docs.microsoft.com/en-us/azure/architecture/data-science-process/overview> (cited February 27, 2022).
33. Ferlitsch A. Making the machine: the machine learning lifecycle (2019). Available at <https://cloud.google.com/blog/products/ai-machine-learning/making-the-machine-the-machine-learning-lifecycle> (cited February 27, 2022).
34. Maskey M, Molthan A, Hain C, Ramachandran R, Gurung I, Freitag B, et al. Machine learning lifecycle for earth science application: a practical insight into production deployment. *Int Geosci Remote Sens Symp.* (2019):10043–6. doi: 10.1109/IGARSS.2019.8899031
35. Wan Z, Xia X, Lo D, Murphy GC. How does machine learning change software development practices? *IEEE Trans Softw Eng.* (2021) 47(9):1857–71. doi: 10.1109/TSE.2019.2937083
36. Shelley K, Cannesson M. Off-label use of clinical monitors: what happens when new physiologic understanding meets state-of-the-art technology. *Anesth Analg.* (2014) 119(6):1241–2. doi: 10.1213/ANE.0000000000000479
37. Antoniou T, Mamdani M. Evaluation of machine learning solutions in medicine. *CMJA.* (2021) 193(36):1425–9. doi: 10.1503/cmaj.210036
38. Karches KE. Against the iDoctor: why artificial intelligence should not replace physician judgment. *Theor Med Bioeth.* (2018) 39(2):91–110. doi: 10.1007/s11017-018-9442-3
39. He J, Baxter SL, Xu J, Xu J, Zhou X, Zhang K. The practical implementation of artificial intelligence technologies in medicine. *Nat Med.* (2019) 25(1):30–6. doi: 10.1038/s41591-018-0307-0
40. Hee Lee D, Yoon SN. Application of artificial intelligence-based technologies in the healthcare industry: Opportunities and challenges. *Int J Environ Res Public Health.* (2021) 18(1):1–18. doi: 10.3390/ijerph18010271
41. Aung YYM, Wong DCS, Ting DSW. The promise of artificial intelligence: a review of the opportunities and challenges of artificial intelligence in healthcare. *Br Med Bull.* (2021) 139(1):4–15. Available at <https://academic.oup.com/bmb/article/139/1/4/6353269> doi: 10.1093/bmb/ldab016
42. Wu E, Wu K, Daneshjou R, Ouyang D, Ho DE, Zou J. How medical AI devices are evaluated: limitations and recommendations from an analysis of FDA approvals. *Nat Med.* (2021) 27(4):582–4. doi: 10.1038/s41591-021-01312-x
43. Wu E, Wu K, Daneshjou R, Ouyang D, Ho DE, Zou J. How medical AI devices are evaluated: limitations and recommendations from an analysis of FDA approvals. *Nat Med.* (2021) 27(4):582–4. doi: 10.1038/s41591-021-01312-x
44. Young AT, Amara D, Bhattacharya A, Wei ML. Patient and general public attitudes towards clinical artificial intelligence: a mixed methods systematic review. *Lancet Digit Heal.* (2021) 3(9):e599–611. doi: 10.1016/S2589-7500(21)00132-1
45. Hripcsak G, Albers DJ. Next-generation phenotyping of electronic health records. *J Am Med Informatics Assoc.* (2013) 20(1):117–21. doi: 10.1136/amiajnl-2012-001145
46. Demiris G, Oliver DP, Washington KT. Defining and analyzing the problem. *Behav Interv Res Hosp Palliat Care.* (2019):27–39. doi: 10.1016/B978-0-12-814449-7.00003-X
47. Zhu L, Zheng WJ. Informatics, data science, and artificial intelligence. *JAMA.* (2018) 320(11):1103. doi: 10.1001/jama.2018.8211