IMDRF International Medical Device Regulators Forum

Draft

IMDRF/SaMD WG/N81 DRAFT: 2024

Medical Device Software: Considerations for Device and Risk Characterization

AUTHORING GROUP

IMDRF Software as a Medical Device Working Group

29 January 2024

Preface

Jeffrey Shuren, IMDRF Chair

Contents

1. Introduction

Software plays an increasingly critical role in healthcare, with a wide range of products serving a variety of medical and administrative purposes in a range of clinical or private settings. A subset of software that is used in healthcare is regulated as a medical device globally by regulatory authorities.

In 2013 the International Medical Devices Regulators Forum (IMDRF) introduced the concept of Software as a Medical Device (SaMD) and subsequently proposed a possible risk categorization framework (IMDRF/SaMD WG/N12 FINAL:2014). Building on the collective experience of its members, the IMDRF SaMD WG now has an opportunity to add to those initial concepts by providing guidance related to device characterization and risk characterization, for a broadened scope of medical device software.

The term “SaMD” has evolved to include a more diverse landscape of software and varied interpretations across jurisdictions. The concepts presented in this document are not exclusive to any specific interpretation of the term SaMD, rather can be helpful to consider more broadly for any software that meets the definition of a medical device.

In this document we refer to this relevant set of software as “medical device software” as a shorthand for document useability. This complex collection of software includes various intersecting and distinct subsets, for example:

- Software that is intended to generate information for use in achieving one or more medical purpose;
- Software that is part of a hardware medical device;
- Software that is not part of a hardware medical device and independent of other medical devices);
- Software that is necessary for a hardware medical device to achieve its intended use/intended purpose;
- Software that is driven or influenced by another medical device;
- Software with an output intended for a human user, medical device, and/or non-medical device;
- Software that uses inputs from humans, medical devices, and/or products that are not medical devices.

Medical device software can operate in complex socio-technical environments—consisting of software, hardware, information technology networks, and people—which form a complex and dynamic interaction between the software function, its inputs and outputs, the intended user, and the unique healthcare circumstances in which the software is used. This complexity together with the interconnectedness of systems, the need for cybersecurity, the speed and frequency of development cycles, the speed at which a solution can be scaled up, and the various aspects of change implementation contribute to the accurate depiction of a device and/or its risk-profile. Medical device software can pose risks that are distinct and unique, such as those that relate to the information that is generated and output by the device and the capacity for varied degrees of clinical autonomy. These devices may be used independently or as part of a platform and span a wide spectrum of risk profiles depending on the intended use, and potential harms associated with use and/or erroneous outputs.

The clear and accurate characterization of a medical device software is fundamental and supports device quality, risk management, regulatory decision-making and device use in healthcare. Stakeholders (including manufacturers, regulators, healthcare providers, end-users, and patients) need to understand what a medical device software is, its purpose, its context of use, how it works and how it changes due to updates. This information can be necessary for proper use and to identify and evaluate the associated hazards, direct and indirect harms, risks and benefits, and to determine device risk classifications.

Risk-based device classifications, applied in accordance with each jurisdiction's regulations, assign the appropriate regulatory obligations in each jurisdiction. Assigning risk categories to these devices can be challenging due to the broad range of technologies and characteristics that can influence risk, the variety of terminology and interpretations used to describe and qualify these devices, as well as the range of classification systems across global regulatory jurisdictions. This document identifies common considerations regarding device characterization and risk characterization to provide a harmonized lens and common language for improved transparency and consistency between stakeholders. This work can help support comprehensive descriptions of medical device software, thorough risk assessments for those devices, as well as interpretations of jurisdictions' classification approaches for these products.

2. Purpose and Scope

2.1. Purpose of the document

The objective of this document is to promote and inform clear and accurate characterizations of medical device software (including intended use/intended purpose statements) and introduce a general strategy for characterizing software-specific risks that leverages the key features of a comprehensive medical device software characterization.

This document is intended to:

- Highlight the importance of comprehensive characterizations for medical device software;
- Establish key features of and common vocabulary for the characterization of medical device software;
- Identify fundamental elements of an intended use/intended purpose statement for medical device software;
- Establish links between characterization features and risk for medical device software;
- Provide information for consideration during the identification and assessment of medical device software risks.

2.2. Scope of the document

This document applies to the subset of software that meets the definition of a medical device (referred to throughout as *medical device software)*, including software that meets the definition of Software as a Medical Device (SaMD) as is defined in the document, *IMDRF SaMD WG N10 Software as a Medical Device: Key Definitions*.

- This document focuses on medical device software irrespective of the software technology and/or the platform (e.g., mobile app, cloud, server, hardware medical device).
- This document is not intended to provide guidance on the regulatory status or classification of products that are not medical devices and provide inputs to software that meets the definition of a medical device.
- This document focuses on software-specific risk considerations and is not intended to be comprehensive of all relevant risk considerations for a medical device software, which may also include additional risks related to interoperable or associated hardware.
- This document is not intended to replace or conflict with existing risk management practices or the development of technical or process standards related to software risk management activities. This document relies on established risk management principles, such as those in *ISO 14971Risk Management for Medical Devices*, in the context of medical device software.

- This document is not intended to replace or conflict with existing IMDRF publications such as those published by the Artificial Intelligence (AI) or Cybersecurity Working Groups; however, it is acknowledged that there are direct relationships and overlap with those publications, and this document is intended to be complementary.
- The content in this document is not regulation and does not imply a convergence of regulations or categorization rules across jurisdictions. Additional work may be required to apply and align these concepts in a given jurisdiction.

3. References

- *IMDRF SaMD WG N10 FINAL:2013 Software as a Medical Device: Key Definitions*
- *IMDRF/GRRP WG/N52:2019 Principles of Labelling for Medical Devices and IVD Medical Devices*
- *GHTF/SG1/N77 Principles of Medical Devices Classification*
- *ISO 14971:2019 Medical Devices - Application of Risk Management to Medical Devices*
- *TIR57: 2016/(R)2023 Principles for medical device security—Risk management*
- *IEC 80001-1:2021 Application of risk management for IT-networks incorporating medical devices*
- *IEC 62304 Medical device software — Software life cycle processes*
- *AAMI TIR57 Principles for medical device security – Risk Management*
- *AAMI TIR34971 Application of Iso 14971 To Machine Learning In Artificial Intelligence—Guide*

4. Device Characterization Considerations

The communication of a comprehensive medical device software characterization (including the intended use/intended purpose and device description) supports stakeholders' ability to understand the device and characterize the associated risks and benefits. This will inform decision-making and help ensure device safety, effectiveness and proper use.

Numerous elements contribute to a comprehensive medical device software characterization, such as the medical purposes, intended users, intended use environment, and intended target populations, as well as the role and timing of the software's use and output in the clinical or healthcare workflow. The characterization should clearly describe what the device is and is intended to do, as well as how, where, when and by whom the software is intended to be used and modified.

This information is essential for identifying and validating the relevant user and clinical requirements, assessing the adequacy of supporting evidence, identifying and controlling risks, determining user-centered labelling and transparency requirements, managing product changes, ensuring proper use while mitigating against misuse, and enabling patient-centered healthcare.

The following two sub-sections discuss considerations for manufacturers when characterizing medical device software within the *intended use/intended purpose statement* and *device description*. These considerations can support the determination of the pertinent and meaningful information to include within medical device software documentation, regulatory submissions, device labelling and user interfaces. All features and attributes listed may not be relevant for every device but are included for consideration. What is communicated will be dependent on the stakeholder and the characteristics determined to have an impact on risk for the specific device.

4.1. Intended Use/Intended Purpose Statement

The intended use/purpose is defined within the *GHTF/SG1/N77 Principles of Medical Devices Classification* document as *the objective intent of the manufacturer regarding the use of a product, process or service as reflected in the specifications, instructions and information provided by the manufacturer*.

The concept of an *intended use/purpose statement* is familiar in many jurisdictions and is typically expected to appear within the device labelling to capture the intended device function and medical purpose, including the indicated diseases, conditions and / or circumstances for which the device is intended to be used. Such statements are generally most useful when they are sufficiently specific and avoid excessively general and/or open-ended language. It is acknowledged that the intended device function and indicated diseases may be considered separate in certain jurisdictions. However, for the purposes of this document, both are relevant and are suggested to be clearly described. For some devices, certain information contained in the sample intended use/intended purpose statement may be included elsewhere in the medical device software labelling, as appropriate.

In order to foster and encourage clear and comprehensive intended use statements for medical device software, *Key Elements* of an intended use/intended purpose statement are captured in section 4.1.1 below. A sample statement guide can be found in Appendix A. It is important to note that not all elements will be applicable to every medical device software and the information provided in these sections is solely for consideration by manufacturers in the development of the medical device software labelling, documentation and regulatory submissions, as appropriate. The sample statement may not be appropriate for all medical device software depending on the technology and intended use.

4.1.1. *Key Elements of Intended Use/Intended Purpose Statement*

1. Medical Purposes
2. Intended Conditions/Diseases/Disorders and Grade/Stage/Level
3. Intended Patient Populations
4. Intended Users
5. Intended Use Environment
6. Contraindications
7. Medical device software function, including:
    - Medical device software inputs
    - Medical device software outputs
    - Explanation of how the medical device software inputs and outputs fit into the clinical or healthcare workflow

4.2. Device Description - Medical Device Software

A detailed medical device software description, accompanying the intended use/purpose statement, is often needed to ensure the comprehensive and adequate communication of all necessary characteristics and information related to a medical device software.

The following four subsections discuss detailed and interrelated information that can be relevant to the characterization of a medical device software, organized according to the following four types or categories of information:

- Medical problem and/or objective
- Context of use
- Function and/or use
- Change management

The information within each category is presented in the form of characterization features with attributes. This non-exhaustive set of considerations for manufacturers is intended to highlight and clarify some important aspects when characterizing a medical device software. The features and attributes within each subsection are tabulated proceeding the discussion; the full set of features and attributes are provided in a summary table in Appendix B.

4.2.1. *Medical Problem and/or Objective*

The medical problem or objective a given medical device software is used for solving or addressing is an important piece of the overall device characterization. This feature can be further broken down into the specific medical purpose, the intended conditions/diseases/disorders, and the intended patient population.

A medical device may be used in different stages of the care pathway, such as diagnosis (e.g., primary diagnosis, screening, triage, staging, etc.); treatment (e.g., relieving symptoms or restoring function); prevention (e.g., averting the occurrence of a disease or condition); prediction (e.g., disease prognosis, anticipated treatment response, etc.) or monitoring (e.g., ongoing assessment of patient parameters). Understanding the specific medical purpose that the device performs or is used in achieving is a key part of characterizing the medical device software.

The condition or disease for which the medical device software is meant to be applied, and the general state of that condition or disease (for example, the grade, stage or level), are important pieces of information at the center of characterizing a medical device software and determining the associated criticality or seriousness of the situation and importance of the output.

Finally, the intended patient population provides an important boundary within which the medical device software is meant to apply and another defining feature of the medical device software characterization. In this document, the term *patient* is used to refer to individuals that receive or await healthcare with the use of the medical device software. The intended patients may be in a *specific subgroup of the population* (e.g., specific age, sex, gender, ethnicity, race, disability, diagnosis; or a fragile and/or vulnerable group; etc.), or *specific intersection of subgroups of the population* (e.g., specific age group + specific sex + those at risk of a specific condition)

The following table summarizes the identified features and attributes that help characterize the medical problem and/or objective.

Table 1 Features and attributes for the characterization of the medical problem and/or objective

Characterization Feature	Potential Feature Attributes
Medical Purpose	**Diagnosis** (e.g., primary diagnosis, screening, triage, etc.), **Prevention**, **Monitoring**, **Mitigation**, **Prediction**, **Treatment**, etc.
Intended Conditions/Diseases/ Disorders and Grade/Stage/Level	**Critical**, **Serious**, **Non-Serious condition or disease,** including consideration of level of progression/stage/ grade (e.g., a chronic condition or an acute change in a chronic condition)
Intended Patient Population	**General population**, **Specific subgroup of the population** (e.g., fragile and/or vulnerable subgroup; specific age group, sex, gender, ethnicity, race, disability, diagnosis, etc.), or

Specific intersection of subgroups of the population (e.g., specific age group + specific sex + those at risk of a specific condition)

4.2.2. *Context of Medical Device Software Use*

The characterization of medical device software extends beyond the device, into the intended circumstances and setting for medical device software use. Two otherwise identical products with different intended contexts of use are distinct devices with different medical device software characterizations. Aspects of that context of use include the *intended user* of the medical device software as well as the *intended use environment*.

The intended user could be a non-clinical user, a non-physician medical professional, a general practitioner medical doctor (MD), a specialist physician, or a combination of these users. A non-clinical user, or lay-user, includes those users that are not trained or qualified to provide medical care, which might include a caregiver or patient user, or other users without medical qualifications. Licensed medical professionals that are non-physicians include nurses, dentists, psychologists, radiation therapists, physiotherapists, etc. General practitioner (GP) medical doctors include, for example, primary care physicians or family doctors, while specialist physicians include radiologists, oncologists, dermatologists, psychiatrists, pathologists, surgeons, etc.

The intended use environment describes the setting in which patient healthcare, with the medical device software, is meant to take place. This could be a non-clinical environment, a general healthcare environment, or a specialty healthcare environment. A non-clinical environment would include home-use; general healthcare environments would include primary care clinics, dental offices, etc.; and a specialty healthcare environment would include, for example, emergency rooms, intensive care units, dermatology clinics, surgical operating rooms, and oncology departments within a hospital.

Another important aspect of the context of use is the finality of the software output and/or its weight in relation to the outcome of the healthcare task/intervention. The *timing within the healthcare task/intervention* is a feature that helps to contextualize the output in terms of being early, midway, or late in the healthcare task/intervention. Similarly, the *role of the software output within the healthcare task/intervention* illustrates the relationship of the output amongst the steps in the healthcare task/intervention, in terms of relative chronology and the software's dependence on and/or input to the other steps. Taken together, these two features help to describe the impact or influence a software may have on the overall trajectory and outcome of a patient's care. These are important to understand the "weight" of the software's use and can help to identify where and how effects from the software use can alter the course of a patient's healthcare experience.

The following table summarizes the identified features and attributes that can help characterize the context of use.

277 278 *Table 2 Features and attributes for the characterization of medical device software context of use*

Characterization Feature	Potential Feature Attributes
Intended User	**Lay user/nonclinical user** (e.g., caregiver, patient user, user without medical qualifications), **Licenced medical professional, non-physician** (e.g., registered nurse, dentist, psychologist, radiation therapist, physiotherapist, etc.), **General Practitioner** (e.g., Primary care physician, family doctor, registered nurse practitioner), **Specialist Healthcare Physician** (e.g., radiologist, oncologist, dermatologist, pathologist, surgeon, etc.)
Intended Use Environment	**Non-clinical Environment** (e.g., home-use), **General Healthcare Environment** (e.g., primary care clinic, virtual primary healthcare), **Specialty Healthcare Environment** (e.g., hospital, specialty clinic, virtual specialty healthcare)
Timing Within Healthcare Task/Intervention	**Early** (e.g., triage, prediction of future diagnoses, early investigations upon suspicious symptoms or information, physiological signal or medical image acquisition for use in diagnosis or treatment planning), **Midway** (e.g., signal or image segmentation for use in diagnosis or treatment planning; routine monitoring of patient health for clinically relevant changes requiring further care and not including acute scenarios), **Late** (e.g., optimal image-guided treatment plan or dosage for consideration; adjunct diagnostic recommendations or second checks; continuous glucose monitor output analysis automatically driving basal insulin dosage; image-guided instrument control in robotic surgery; autonomous detection and diagnosis of diabetic retinopathy) *** Note:** these 3 phases (Early, Midway and Late) described above serve as reference points, and it is not crucial to state which phase should be applied. Rather, it is important to characterize the timing of the output relative to the final intervention, decision, or action as well as the relative chronology of how the product will be introduced in relation to other steps (e.g., prior steps, concurrent steps, conditional steps, subsequent steps) and current standard medical practices.
Role Of Software Output Within the Healthcare Task/Intervention	**Software output's relationship to the healthcare task/intervention steps**, such as the output's contribution to the relevant healthcare decision or action (for example, intended as an aid that is combined with current practice); alteration of standard/current practice (for example, intended to replace or substitute all or part of current practice, to provide a new scheme, etc.); dependence on other steps (e.g., uses output values or clinical decisions from prior steps, concurrent steps, conditional steps); and/or influence over other steps (e.g., provides input to concurrent steps, subsequent steps, conditional steps, or final intervention/decision).

4.2.3. *Medical Device Software Function and/or Use*

The function and use of a medical device software can be described by various aspects, such as the generation of outputs, the output itself and how that output fits into the care pathway.

The types of output provided by a medical device software could be a *clinical interpretation or intervention*, a *workflow recommendation*, or *data or information* for use in a medical purpose. *Clinical interpretations or interventions* can include, for example, a probability, prediction, detection, diagnosis, severity, prognosis, grade, or stage of a disease or condition; or the prescription, treatment, therapy, recommended dosage or treatment plan for a disease or condition. A *workflow recommendation*, in contrast, is not an interpretation on the clinical decision or action but rather an intermediate step in the workflow, such as recommended contrast dye dosage; imaging technique, modality, or parameters; surgical tool choice; supplementary medical tests, etc. *Data for use in medical purpose* is output by a medical device software for use in a medical purpose and is typically more objective, such as anatomy measurements or volumes, segmented or contoured organs, tissues; processed, reconstructed, or de-noised images; processed signals or waveforms such as from electrocardiographs or electroencephalographs.

The input to the medical device software influences the function of the device and is fundamental to understanding the medical device software, the output, and the associated risks and considerations. The source of those inputs may be a *human user* (e.g., patient inputted symptoms or conversations), a *medical device* (e.g., a medical image), or a non-medical device or *consumer product* (e.g., smart-phone photos, EHR data from patient chart). Notably, the inputs to a medical device software do not necessarily have to be medical information or to come from a medical device. Regulators may consider the impact that non-medical data or data sources have on the safety and effectiveness of a medical device software. However, the use of non-medical data sources in a medical device software does not change the regulatory status of the source of non-medical data.

The level of automation of the task and output refers to the degree to which the output requires and receives review and approval by the user, which can range from fully automated or conditionally automated to semi-automated, and manual. A *fully automated* output does not require review or approval and cannot be modified by the user, while *conditionally automated* tasks have some outputs that are flagged for review and the user has a way to go back and edit the output, for example if it is assigned low confidence or high risk. Semi-automated outputs are made available for critical assessment and approval or editing and, finally, for manual outputs, the user controls the generation of the output. The level of automation is determined irrespective of whether the user is a clinical or non-clinical user.

The degree of clinical autonomy is a spectrum of capacities or liberties to operate independently of a clinical user's guidance. An autonomous device provides outputs that impact the subsequent clinical action or decision without a clinical user in the loop (for example, with no user in the loop or a non-clinical user in the loop). Conditionally autonomous outputs will meet this condition selectively (for example, for certain results, input characteristics, or circumstances). Supervised outputs can impact subsequent clinical actions or decisions without a clinical user having to approve the output but operate under the supervision of adequately qualified clinical intended users. Non-autonomous outputs are typically intended to augment, assist, or inform a clinical user in their determination of a clinical decision or action.

The level of intelligibility, transparency, and explainability of the underlying logic is also an important characteristic of a medical device software. This includes the information about the software algorithm or technology utilized (such as, deterministic formulae; machine learning approaches; mathematical simulations; etc.) and information about how an output or result was reached or the basis for a decision or action. This aspect could be attributed as *explained and comprehensible*; *partially explained or partially evaluable* (e.g., output provided with saliency maps); or as *not explained or incomprehensible* (e.g., Black Box). Understanding this aspect of a medical device software contributes to the assessment of risks and uncertainties, as well as determining supporting evidence expectations.

The destination or target of the output could include outputs intended to be provided to *human users*, or to *medical devices* or *consumer products* (either with or without intermediate use by a human user).

The following table summarizes the identified features and attributes that can help characterize the device function and/or use.

Table 3 Features and attributes for the characterization of medical device software function and/or use

Characterization Feature	Potential Feature Attributes
Output Type	**Clinical Interpretation or Intervention** (e.g., diagnosis, suspicion, probability, prediction, detection, severity, prognosis, grade, stage, direct markers of a diagnosis, prescription, treatment/therapy, recommended treatment, recommended dosage, radiation treatment plan), **Workflow Recommendation** (e.g., contrast dye dosage; recommended imaging technique/modality/parameters; recommended surgical tool choice; recommended additional test based on established guidelines), **Data for use in medical purpose** (e.g., anatomy measurement, volume, or segmentation; processed image/image reconstruction/de-noised image; processed signal/waveform (e.g., processed ECG))
Input Source	From **human user**, **medical device**, or **consumer product**
Level of Task Automation	**Fully automated** (i.e., output does not require review/approval and cannot be modified by the user), **Conditionally automatic** (some outputs are flagged for review or user has a way to go back and edit the output, for example if assigned low confidence/high risk), **Semi-automatic** (processed output is made available for critical assessment and approval or editing), **Manual** (user controls generation of output)
Degree of Clinical Autonomy	**Independent/Autonomous** (i.e., output impacts subsequent clinical action or decision without clinical user in the loop), **Conditionally independent/ autonomous** (output selectively impacts subsequent clinical action or decision without clinical user in the loop; this can include medical device software that require non-clinical user screening decisions),

	Supervised (i.e., output impacts subsequent clinical action or decision without clinical user having to approve, but with supervision from adequately qualified operator), **Non-autonomous** (output augments/ assists/ informs clinical user in their determination of clinical decision/action)
Intelligibility/Transparency/Explainability (Underlying Logic including the Algorithm/Technology used and How an Output is Reached)	**Output is not explained or cannot be understood** (e.g., Black Box), **Output is partially explained or can be partially evaluated** (e.g., output provided with saliency maps), **Output is explained and can be comprehended**
Destination/Target of Output	Input to **human user**; Input to **medical device**; Input to a **consumer product**

4.2.4. *Medical Device Software Change Management*

The change management approaches tied to a device form part of the device characterization, including the autonomy of learning or change implementation as well as the intended domain of change implementation.

The degree of learning or change management autonomy describes the effectuation and control of training, learning and updates to the medical device software. Possible attributes within this feature can include *self-learning* (autonomous updates effectuated and controlled from within medical device software) and *externally controlled learning* (non-autonomous updates effectuated and controlled by manufacturer and/or user).

The domain of learning or change implementation refers to the scope or applicable extent of change. This might be described as being applicable on a scale that is *international*, *national*, *regional*, *clinic-specific*, or *patient-specific*.

Another aspect of software change management is the infrastructure for installation, updates and error corrections. Updates and changes to the software can be provided in response to software failures, errors, opportunities for improvement, critical performance updates, and recalls. Software-specific risks and risk controls can depend on the software *distribution channels* (app stores, manufacturer homepage, web application, etc.) and *software installation locations* (mobile phones, hardware medical devices, or personal computers (PCs) of the users, server anywhere in the world or one single server at the manufacturer site).

Distribution channels, such as app stores offering medical device software, may not be regulated in all jurisdictions. Surveillance challenges and unclear responsibilities may occur in cases of recalls, field safety corrective actions and distribution of information. Furthermore, software *installation location* can influence the effectiveness and speed of access to updates or the deactivation of erroneous or recalled software and the traceability of affected installations and users.

The following table summarizes the identified features and attributes that can help characterize the device change management, including the degree of change autonomy, the change domain and infrastructure for installation, updates and error correction.

Table 4 Features and attributes for the characterization of medical device software change management

Characterization Feature	Potential Feature Attributes
Degree of Learning/Change Management Autonomy	**Self-learning/autonomous learning** (autonomous updates effectuated and controlled from within medical device software), **Externally controlled user-driven learning/change** (non-autonomous updates effectuated and controlled by the user), **Externally controlled manufacturer-driven learning/change** (non-autonomous updates effectuated and controlled by the manufacturer)
Domain of Learning/Change Implementation	**International, National, Regional, Clinic/Site-specific, Patient-specific**
Installation, Update and Error Correction Infrastructure	**Distribution channels** (e.g., app stores, manufacturer homepage, web application), **Installation locations** (Mobile phones, hardware medical devices, or PCs of the users, server anywhere in the world or one single server at the manufacturer site

5. Medical Device Software Risk Characterization

Identifying and estimating medical device software-specific risk can raise unique questions compared to other medical devices. Risk management approaches, such as those proposed within ISO 14971, often describe risk as the combination of the probability of occurrence of harm and the severity of harm. Harms, however, can be both direct and indirect, and a comprehensive identification of software-specific contributions to possible harms can be challenging because software, *on its own*, does not pose "physical" hazards to which harms can be easily attributed. Evaluating software-specific contributions to possible harms generally requires interpretation of primarily performance-related hazards[1], or more specifically information-related hazards, and understanding the associated risk is then critically tied to a complete understanding of a device's intended use/purpose and particular implementation.

In other words, when assessing the risk of medical device software, it is important to understand the contribution of information-related hazardous situations, which are closely tied to the role of software in achieving an intended medical purpose. These hazardous situations can generally be understood through the lens of "performance-related hazards," as described in ISO 14971, such as hazards relating to data access, availability, delivery, and diagnostic information as opposed to, for example, energy, biological, or chemical hazards.

An accurate characterization of software, including its characteristics such as intended use, output type, use environment, autonomy, etc., allows for both a more comprehensive identification of these direct and indirect harms and a clear understanding of how software-specific harms can then lead to risks unique to a given intended use/purpose.

While the performance-related hazards and risks related to software do not always account for the *totality* of risk posed by a device (such as in the case of software that may supply data or generate the inputs for a hardware actuator that poses associated physical hazards), it is important to fully characterize the impact of a particular software implementation or solution on device risk because it can still lead to demonstrable impacts on patient safety or device effectiveness through direct or downstream means.

[1] ISO 14971:2019 Medical Devices – Application of Risk Management to Medical Devices

Further, it is important to consider that software-specific hazards often sit at the junction of both safety and cybersecurity risks. Therefore, it can be helpful to consider software-specific considerations pertaining to harm as a combination of how harm is defined for safety and cybersecurity. In other words, medical device software-specific consideration of harm could be viewed as relating to injury or damage to the health of people[2] *and reduction of effectiveness*[3] – where "reduction of effectiveness" can result from inadequate, incorrect, or absent data supplied to a human or product at an inappropriate time, rate, or with an inadequate method. For example, injection of unwanted or unintended bias into a decision-making system, whether or not it results in direct harm to a patient, can be understood as a harmful reduction in effectiveness. In other words, the introduction of the particular software solution has had a negative impact on the decision-making system. Often, this can also be viewed as accounting for "indirect harm" from the software, as noted above.

Performance-related hazards pertinent to software – that is, specifically information-related hazards - can impact the function of other products or systems, how workflows or processes are informed, and can directly impact user decision making. As such, a harmonized discussion of how to identify, characterize, and contextualize these possible harms and their impact on device risk can provide greater understanding for why risk categorization for medical device software may be highly variable across regulatory jurisdictions, as well as how to articulate these differences more consistently.

Key Points:

- When evaluating the risk posed by software, both direct and indirect harms should be considered.
- Because hazards associated with software are typically information-based hazards (such as delayed, inappropriate, or erroneous information), it is important to consider potential harm as both injury or damage to health as well as a reduction in effectiveness when accounting for indirect harms.
- The possible harms and associated risks related to implementing software are dependent on a device's specific intended use.

Below, general considerations for identification and analysis of software-specific hazardous situations are discussed, as well as considerations when carrying forward these hazardous situations as part of risk estimation. These approaches are intended to provide a shared means of discussing the unique risks posed by software that meets the definition of a medical device, and how such an understanding may drive device risk categorization across any number of risk categorization systems, layers in part or in whole.

[2] While ISO 14971:2019 defines harm as "injury or damage to the health of people, or damage to property or the environment." it can be helpful to consider, more specifically, harm as it relates to "injury or damage to the health of people" when discussing medical device safety in this document. The narrower definition of patient harm has the net effect of prioritizing regulatory review of those changes necessary to protect public health.

[3] Harm is defined in TIR57: 2016/(R)2023 as "physical injury or damage to the health of people, or damage to property or the environment, or reduction in effectiveness, or breach of data and systems security" as described in IEC 80001-1:2021.

5.1. Identification and Analysis

The success of risk assessment and management activities hinges on the risk assessors' understanding of what the medical device software is and is meant to do, as well as how, where, when and by whom the medical device software is meant to be used. The comprehensive characterization of medical device software, considering the information presented in section 4 of this document, provides the foundation necessary for software-specific risk characterization. Approaches to identifying and considering risks within each of the information groupings in section 4 are provided below, in part, to illustrate the way many variables contribute and interact to form a more complete understanding of the unique risks that may impact a particular medical device software.

To identify and characterize software risk, it is helpful to step through the process of first identifying a device characteristic, then asking *why* the characteristic matters to the intended use/purpose of the software, and then identifying the hazardous situations that may arise based upon both the intentional software design decisions and unintentional software failures. It remains important, however, to ensure that exploring device characteristics in this manner is not done in a vacuum and interdependencies of the software are carefully considered to comprehensively describe a medical device software's "risk characterization."

Appendix C provides questions for consideration to accompany each characterization feature previously identified in section 4. These questions are provided to help develop an understanding of "why the characteristic matters to the intended use/purpose of the software," as a means to helping to identify specific hazardous situations that may be related to the software's design and intended use/purpose. While not comprehensive, the questions aim to highlight how the context provided by each of a device's unique characteristics could impact an understanding of the potential harms introduced by a particular software, and thereby affect the overall risk of the medical device. The questions are intended to help guide a thorough consideration of potential harms a medical device software could introduce, and not all questions may be applicable or relevant to every medical device software.

Appendix D includes examples illustrating how answering the questions in Appendix C can help to surface the way different characterization features and their interactions may affect an understanding of the risks introduced by a particular medical device software. Importantly, identifying these "software-specific" contributions to device risk is intended as a means of articulating why the *software* for a particular medical use/purpose may or may not alter device risk categorization under any number of frameworks. This concept is discussed further in section 5.3 of this document.

5.2. Estimation

As noted above, risk management approaches, such as those proposed within ISO 14971, often describe risk as the combination of the probability of occurrence of harm and the severity of harm. These risk estimation features, together with medical device software characterization features outlined in section 4 of this document, can be essential in assessing and managing risks.

For medical device software, this determination requires the identification of the potential *direct and indirect* harms associated with hazardous situations, such as erroneous outputs from the software, followed by an assessment of the severity of those harms, such as reductions in life expectancy, psychological injury, or inappropriate or unnecessary invasive treatment. While probability of harm can generally be helpful to consider when estimating risk, there is not broad consensus on a method for quantitatively estimating probability of occurrence of software failure. Additionally, cybersecurity risk management often considers exploitability of vulnerabilities rather than probability of occurrence of harm; and it is generally understood that probability of software-related harms can be influenced by factors like usability, which can make estimation further challenging.[4] To this end, when estimating software-specific risks, it can be helpful to set the probability of software failure to 1 and if possible, estimate the probability based on other factors to perform risk estimation.

The guiding questions in section 5.1 can provide a basis for isolating software-specific hazards *and* for contextualizing their potential severity of harm, on the basis of understanding how applying a specific software solution can affect the way the medical device intended use/purpose is achieved.

These concepts can then be leveraged for risk characterization (e.g., through risk assessment per ISO 14971) and the determination of the severity of direct or indirect harm caused by a given, software-specific hazardous situation (e.g., catastrophic, critical, serious, minor, negligible). Once harms are identified, the approach to "software-contributed" risk estimation is not unique. That said, when software may need to be considered in the context of a broader device to achieve an intended use/purpose, it can also be helpful to consider whether the software becomes a single-point failure for a given possible harm and, if so, how this may impact risk estimation and associated mitigations.

5.3. Approaches for Risk Categorization

It may not be universally possible or beneficial to create completely rigid and distinct categories of risk for any one type of function, disease, intervention, population, or user. For any given medical device software there may be both interdependencies and unequal weight amongst characterization groupings that ultimately inform the understanding of device risk and, therefore, may impact a subsequent categorization. Further, when addressing the specific contribution to device risk posed by software, considerations like how supplied information will be ingested by a given userbase, as just one example, may reasonably not have uniform or universal answers across jurisdictions.

[4] Ref: IEC 62304, AAMI TIR57, AAMI TIR34971

Different jurisdictional authorities may have distinct philosophies and legal obligations which shape their different risk-based classifications. Therefore, the discussion provided in section 5 and further illustrated in Appendix B is intended as a common basis for considering and articulating how characterization features impact the risk of software that meets the definition of a medical device, particularly through the lens of the interdependent factors shaping an understanding of risk specific to software for a given intended use/purpose. Put another way, this document intends to provide insight into *how* a particular software risk categorization could be concluded without prescribing a single “correct” and universal category to any given device. As noted previously, among other complications, software-specific risks may have a significant but not exclusive influence on the risk categorization applicable to a given device. In premise, this document can serve as the basis for discussing a given understanding of a medical device software’s risk within a broader device system or regulatory structure.

6. Considerations for Implementation

Harmonizing approaches to the characterization of medical device software will support the assessment of device risks and benefits for all stakeholders. Providing a common basis for describing these devices and considering how different characteristics impact risk can help promote safety and effectiveness as well as consistency and alignment across jurisdictions.

The considerations presented in Sections 5.0 and 6.0 can be used to support understanding of a medical device software and its risks and facilitate the interpretation and application of different device risk classification systems across jurisdictions.

Device classification in a given jurisdiction will ultimately be dictated by the governing authorities, laws, and regulations. To the extent possible, jurisdictions may consider incorporating harmonized language and concepts from this document into their local guidance or processes, for example, connecting the device and risk characterization language in the document to their labelling and risk management expectations or classification regulations.

Jurisdictions may be able to leverage a subset of characterization features and attributes, together with the assessment of medical device software risks and their severity, to describe their approach to applying risk categorization to medical device software.

These concepts are intended to be used by stakeholders alongside their existing frameworks, to provide additional detail and exposition for decision-making – ultimately promoting and informing clear, consistent, and accurate characterizations of medical device software.

Appendix A: Sample Intended Use/Intended Purpose Statement

In order to foster and encourage clear and comprehensive intended use statements for medical device software, *Key Elements* of an intended use/intended purpose statement are captured in section 4.1.1. A sample statement guide can be found below. It is important to note that not all elements will be applicable to every medical device software and the information provided in these sections is solely for consideration by manufacturers in the development of the medical device software labelling, documentation, and regulatory submissions, as appropriate. The sample statement may not be appropriate for all medical device software depending on the technology and intended use. Although typically included in the intended use/ intended purpose statement, for some devices, information such as contraindications, may be included elsewhere in the medical device software labelling due to the volume of information.

> The *[name of medical device software]* is software intended for use in the *[medical purposes]* of *[conditions/diseases/disorders]* in *[intended patient populations]*. This software is intended to be used by [*intended user populations]* in *[intended use environments]*. This medical device software is contraindicated for *[contraindications]*. This medical device software uses *[inputs]* in order to produce *[description of outputs]*. These outputs are *[description of how the output is intended to be used, how it fits in the clinical or healthcare workflow and how it contributes to the final healthcare decision/action*].

Appendix B: Characterization Feature Summary Table

Information Grouping	Characterization Feature	Potential Feature Attributes
Medical Problem and/or Objective	Medical Purpose	**Diagnosis** (e.g., primary diagnosis, screening, triage, etc.), **Prevention**, **Monitoring**, **Mitigation**, **Prediction**, **Treatment**, etc.
	Intended Conditions/Diseases/ Disorders and Grade/Stage/Level	**Critical**, **Serious**, **Non-Serious condition or disease,** including consideration of level of progression/stage/ grade (e.g., a chronic condition or an acute change in a chronic condition)
	Intended Patient Population	**General population**, **Specific subgroup of the population** (e.g., fragile and/or vulnerable subgroup; specific age group, sex, gender, skin tone, race, disability, diagnosis, etc.), or **Specific intersection of subgroups of the population** (e.g., specific age group + specific sex + those at risk of a specific condition)
Context of Medical Device Software Use	Intended User	**Lay user/nonclinical user** (e.g., caregiver, patient user, user without medical qualifications), **Licenced medical professional, non-physician** (e.g., registered nurse, dentist, psychologist, radiation therapist, physiotherapist, etc.), **General Practitioner** (e.g., Primary care physician, family doctor, registered nurse practitioner), **Specialist Healthcare Physician** (e.g., radiologist, oncologist, dermatologist, pathologist, surgeon, etc.)
	Intended Use Environment	**Non-clinical Environment** (e.g., home-use), **General Healthcare Environment** (e.g., primary care clinic, virtual primary healthcare), **Specialty Healthcare Environment** (e.g., hospital, specialty clinic, virtual specialty healthcare)

	Timing Within Healthcare Task/Intervention	**Early** (e.g., triage, prediction of future diagnoses, early investigations upon suspicious symptoms or information, physiological signal or medical image acquisition for use in diagnosis or treatment planning), **Midway** (e.g., signal or image segmentation for use in diagnosis or treatment planning; routine monitoring of patient health for clinically relevant changes requiring further care and not including acute scenarios), **Late** (e.g., optimal image-guided treatment plan or dosage for consideration; adjunct diagnostic recommendations or second checks, continuous glucose monitor output analysis automatically driving basal insulin dosage; image-guided instrument control in robotic surgery; autonomous detection and diagnosis of diabetic retinopathy) *** Note:** these 3 phases (Early, Midway and Late) described above serve as reference points, and it is not crucial to state which phase should be applied. Rather, it is important to characterize the timing of the output relative to the final intervention, decision or action as well as the relative chronology of how the product will be introduced in relation to other steps (e.g., prior steps, concurrent steps, conditional steps, subsequent steps) and current standard medical practices.
	Role Of Software Output Within the Healthcare Task/Intervention	**Software output's relationship to the healthcare task/intervention steps**, such as the output's contribution to the relevant healthcare decision or action (for example, intended as an aid that is combined with current practice); alteration of standard/current practice (for example, intended to replace or substitute all or part of current practice, to provide a new scheme, etc.); dependence on other steps (e.g., uses output values or clinical decisions from prior steps, concurrent steps, conditional steps); and/or influence over other steps (e.g., provides input to concurrent steps, subsequent steps, conditional steps, or final intervention/decision).
Medical Device Software Function/ Use	Output Type	**Clinical Interpretation or Intervention** (e.g., diagnosis, suspicion, probability, prediction, detection, severity, prognosis, grade, stage, direct markers of a diagnosis, prescription, treatment/therapy, recommended treatment, recommended dosage, radiation treatment plan), **Workflow Recommendation** (e.g., contrast dye dosage; recommended imaging technique/modality/parameters; recommended surgical tool choice; recommended additional test based on established guidelines), **Data for use in medical purpose** (e.g., anatomy measurement, volume, or segmentation; processed image/image reconstruction/de-noised image; processed signal/waveform (e.g., processed ECG)

	Input Source	From **human user**, **medical device**, or **consumer product.**
	Level of Task Automation	**Fully automated** (i.e., output does not require review/approval and cannot be modified by the user), **Conditionally automatic** (some outputs are flagged for review or user has a way to go back and edit the output, for example if assigned low confidence/high risk), **Semi-automatic** (processed output is made available for critical assessment and approval or editing), **Manual** (user controls generation of output)
	Degree of Clinical Autonomy	**Independent/Autonomous** (i.e., output impacts subsequent clinical action or decision without clinical user in the loop), **Conditionally independent/ autonomous** (output selectively impacts subsequent clinical action or decision without clinical user in the loop; this can include medical device software that require non-clinical user screening decisions), **Supervised** (i.e., output impacts subsequent clinical action or decision without clinical user having to approve, but with supervision from adequately qualified operator), **Non-autonomous** (output augments/ assists/ informs clinical user in their determination of clinical decision/action)
	Intelligibility/Transparency/Explainability of Underlying Logic including the Algorithm/Technology used and How an Output is Reached	**Output is not explained or cannot be understood** (e.g., Black Box), **Output is partially explained or can be partially evaluated** (e.g., output provided with saliency maps), **Output is explained and can be comprehended**
	Destination/Target of Output	Input to **human user,** Input to **medical device,** Input to a **consumer product**
Medical Device Software Change Management	Degree of Learning/Change Management Autonomy	**Self-learning/autonomous learning** (autonomous updates effectuated and controlled from within medical device software), **Externally controlled user-driven learning/change** (non-autonomous updates effectuated and controlled by the user), **Externally controlled manufacturer-driven learning/change** (non-autonomous updates effectuated and controlled by the manufacturer)

	Domain of Learning/Change Implementation	**International, National, Regional, Clinic/Site-specific, Patient-specific**
	Installation, Update and Error Correction Infrastructure	**Distribution channels** (e.g., app stores, manufacturer homepage, web application), **Installation locations** (Mobile phones, hardware medical devices, or PCs of the users, server anywhere in the world or one single server at the manufacturer site)

597

Appendix C: Example Considerations to Understand Software Hazards Associated with Device Design and Intended Use

The questions noted in the below table are intended to help guide a thorough consideration of potential harms that a medical device software could introduce. Not all questions may be applicable or relevant to every medical device software. This is not intended to be an exhaustive or required list of considerations for the intended use or the intended user of the medical device software, rather they are optional examples that may be helpful to consider while characterizing software risk.

Information Grouping	Characterization Feature	Considerations for Medical Device Software Risk Characterization
Medical Problem and/or Objective	Medical Purpose	•Is the medical device software intended to be used as **adjunctive or alongside** other tools or treatment? Is the medical device software **intended to replace or augment** a system or process? If it is meant to augment, in **what manner** is the medical device software augmentative (for example, is the software output **additive or confirmatory** to another process or outcome)? •Is the output of the software, itself, intended to be **therapeutic or a treatment**? Is the software output used for decision making with diagnosis or therapeutic purposes? Is the software used to monitor physiological processes or vital physiological parameters? Does the software have alarm functions used to prompt immediate intervention?
	Intended Condition/Diseases/ Disorders and Grade/Stage/Level	•How, if at all, does the condition/disease (for example, acute or chronic) that the medical device software intended for **impact the criticality** of the data output by the software? •Does the condition/disease **modify the timing of when the information is needed** or **is provided** or must be used? •Does the condition/disease **define the sensitivity or accuracy** of the information needed for the input or output of the software? Could the nature of variation of monitored parameters result in immediate danger to the patient? •Could the decisions or diagnostics made by the software output have an impact that may cause death or an irreversible deterioration in condition/disease or a serious deterioration in condition/disease or a surgical intervention?
	Intended Patient Population	•Does the intended patient population include a specific **vulnerable subgroup**? •How **diverse** is the intended patient population? How generalized does the information need to be to **perform adequately** across the intended patient population? How specific? •Does the medical device software accurately **reflect** the demographics, backgrounds, and characteristics of the population the software will be used for?
Context of Medical Device Software Use	Intended User	•Does the medical device software enable **new/different users** to achieve the clinical task than those who would perform the task without the software? •Does the user need to possess **expertise**, or access to expertise, to understand the inputs and/or outputs of the software?

	Intended Use Environment	•Is use of the medical device software providing a clinical task or service in an **environment that would not otherwise have such tasks or services** available (e.g., would otherwise require an expert present)? •Is the device intended to be used in an uncontrolled or unconventional setting? •Can external factors, both physical and digital, affect the use, input or output of the device? •Do the expected virtual conditions and computing environment require additional software controls and/ or impact the users' access to the software?
	Timing Within Healthcare Task/Intervention	•Does the user have **adequate time to review the basis** for the information output by the software or to review and curate the information being used as input to the software? •Could the software output **initiate a healthcare intervention that would not otherwise be identified** by a particular user or in a particular setting (e.g., pre-screening information prompting a patient to speak to a doctor about a possible condition)? •Are there **possible harms or dangers related to the healthcare task/intervention that could occur immediate** to the software's outputs? •Are there **possible harms or dangers related to the healthcare task/intervention that could occur distantly** from the use of the software, but are related to decision points generated by the software's outputs?
	Role Of Software Output Within the Healthcare Task/Intervention	•Does an erroneous output from the software at the intended point in the workflow **put the patient on a path toward subsequent harm**? •Is the **frequency of output** appropriate to its role and timing in the workflow (e.g., is there a potential for notification fatigue)? •Does the software create a **single point of failure** in the clinical task/intervention?
Device Function/ Use	Output Type	•Is the output **supplementing** additional information to contribute to a clinical interpretation or workflow recommendation? Is it a **replacement or substitution** for information meant to determine a clinical interpretation, workflow recommendation, or as data for use in a medical purpose? •Is the output commonly accepted in **clinical practice** or based upon sound **scientific principles**? Is the output **proprietary**? •Is access to the output **tiered or limited** by user or other credentials?

		•Is the output Boolean, e.g., values that are either true, or false?
	Input Source	•Is the input source from a human user, medical device, or consumer product? •Is the input source **unique** or could the data be obtained through **other methods or sources**? •Is an adequate input source **governed by specific parameters** such as rate, sensitivity, or precision (inclusion and exclusion criteria)? Is the input relevant? •Is the input data direct or **informed or transformed** by other tools, products, or intermediaries? Are the transformed data suitable? •Are there **multiple input sources** or data types? Are they **interdependent**?
	Level of Task Automation	•Does the user **control generation** of the output? •Does the output require **review/approval** and allow **modification** by the user? •Are outputs flagged for review or does the software provide a way to go back and **edit the output**, for example if assigned low confidence/high risk? •Can the user **review the basis** for the output? •Does the user **control or review** the software **inputs**?
	Degree of Clinical Autonomy	•Is a user in the loop? Is the user in the loop a health care professional?
	Intelligibility/Transparency/Explainability of Underlying Logic including the Algorithm/ Technology used and How an Output is Reached	•Is the functionality of the product **sufficiently explained and reasonably understood** by the patient? •Is the functionality of the product **explained and understood** by users other than the patient? Is **different information provided** to different user groups or patients? •Is the functionality **partially explained** or **partially able to be evaluated** by the user? (e.g., output provided with saliency maps)**.**
	Destination/Target of Output	•Is the output **the only instruction/data/information** needed to drive the target's next action?
Medical Device Software Change Management	Degree of Learning/Change Management Autonomy	•Does the medical device software **independently change** its underlying algorithms? •**How often** is medical device software performance verified? •Are updates to algorithmic performance **driven by** non-clinical or clinical users, or manufacturer driven, or a combination of these users?

	Domain of Learning/Change Implementation	• Is **domain-specific implementation** necessary to achieve adequate software performance? • Where are changes intended to be implemented and **how variable are these domains**?
	Installation, Update and Error Correction Infrastructure	•What specific **channels** are used to **distribute** the medical device software? •Does the medical device software have **multiple installation locations**? Where are corrections initiated?

610

Appendix D: Example of Discussing Information Risk in Application to Risk Characterizations

When considering a possible framework for risk categorization, each jurisdiction manages different constraints best addressed by a convergence of risk categorization strategies. This section includes examples applying the considerations described in sections 5 and 6. The examples below are intended to help illustrate how robustly characterizing software and systematically assessing the contribution of characterization factors to the software risk can provide a shared and more granular means of discussing risk that remains transferrable between potentially diverse risk categorization structures.

Below we have provided a full example of a software function applying the considerations discussed in the above sections as well as specific examples highlighting how changes in specific groupings of characterization features may impact risk.

Example A: Software function that serves as a primary diagnostic to identify patients with prediabetes

> **Scenario:** *The software is intended to analyze health related data including data from electronic health records, laboratory tests, and other diagnostic tests to identify individuals with pre-diabetes (i.e., an early marker of diabetes) with an output that is reviewed by healthcare practitioners.*

A.1 Sample Intended Use/Intended Purpose Statement

> The ***Product X*** is software intended for use in the ***diagnosis*** of ***prediabetes*** in ***adult men and women at risk of developing diabetes***. This software is intended to be used by ***medical professionals*** in ***general healthcare environments***. This medical device software is developed using ***a machine learning model***. This medical device software is used for ***patients without an existing diabetes diagnosis***. This medical device software uses ***specific data within the electronic health records (EHR)*** in order to produce a ***conditionally automatic algorithm output that provides likelihood of developing diabetes***. These outputs are ***conditionally independent/ autonomous*** (i.e., output is presented to healthcare providers for review above a threshold %) and are intended to be used as a ***clinical workflow recommendation for additional testing or follow-up based on established guidelines***.

As discussed in Appendix C, addressing each of the characterization features through corresponding questions is helpful for evaluating risk. The below questions are listed by information grouping to support comprehensive discussion of risk considerations.

A.2 Software Risk Considerations:

A.2.1 Medical Problem and/or Objective

Characterization Feature	Considerations for Medical Device Software Risk Characterization
Medical Purpose	•Is the medical device software intended to be used as **adjunctive or alongside** other tools or treatment? Is the medical device software **intended to replace or augment** a system or process? If it is meant to augment, in **what manner** is the medical device software augmentative (for example, is the software output **additive or confirmatory** to another process or outcome)? •Is the output of the software, itself, intended to be **therapeutic or a treatment**? Is the software output used for decision making with diagnosis or therapeutic purposes? Is the software used to monitor physiological processes or vital physiological parameters? Does the software have alarm functions used to prompt immediate intervention?
Intended Conditions/ Diseases/ Disorders and Grade/Stage/Level	•How, if at all, does the condition/disease (for example, acute or chronic) that the medical device software is intended for **impact the criticality** of the data output by the software? •Does the condition/disease **modify the timing of when the information is needed** or is **provided** or must be used? •Does the condition/disease **define the sensitivity or accuracy** of the information needed for the input or output of the software? Could the nature of variation of monitored parameters result in immediate danger to the patient? •Could the decisions or diagnostics made by the software output have an impact that may cause death or an irreversible deterioration of condition/disease or a serious deterioration in condition/disease or a surgical intervention?
Intended Patient Population	•Does the intended patient population include a specific **vulnerable subgroup**? •How **diverse** is the intended patient population? How generalized does the information need to be to **perform adequately** across the intended patient population? How specific? •Does the medical device software **accurately reflect** the demographics, backgrounds, and characteristics of the population the software will be used for?

In this example, we first consider questions related to the Medical Problem and/or Objective:

In considering the *Medical Purpose,* we recognize that this medical device software is intended to be used alongside other tools or treatments i.e., used alongside additional diagnostic test results, treatments, and data available in electronic health records. The medical device software is intended to augment a system or process i.e., the software output is used as a tool to aid in the diagnosis of pre-diabetes. Here, it is helpful to consider that the software is intended to augment and aid, which suggests the output may not be the sole influence on the related clinical decision point. If the software output is not a single point failure that will lead to patient harm, this can impact our understanding of the software's risk.

When considering the *Intended Condition/Disease/Disorders and Grade/Stage/Level* of the patient, we consider that the general state of the condition as a pre-disease state (i.e., the state of a condition before it is a disease) does not impact the criticality of the output of the software. The general state of the condition being a pre-disease state determines that the information is needed or must be used before the disease (diabetes) is diagnosed to predict a high likelihood of subsequently developing the disease (diabetes). Furthermore, the general state of the condition as a pre-disease state (i.e., the state of a condition before it is a disease) and the likelihood of a pre-diabetes state being present (i.e., pre-test probability) determines the sensitivity and/or accuracy of the information needed for the output of the software. Given these factors, the software output is unlikely to have an impact that may cause death or an irreversible deterioration of condition/disease, which can be helpful to consider when evaluating the overall impact that a software failure could have on the device risk. In this case, the risk may be generally lower, because the output's relationship to the condition is not one that may likely lead to irreversible harm.

The *Intended Patient Population* in which this medical device software is intended to be used includes the general public but may include vulnerable subgroups such as individuals of different ethnicities, different age groups (e.g., <40, 40-60, >60 years old). The intended patient population is the general public that is representative of the demographics in the local userbase which may include regional, state, or at the national level. This information needs to be broadly generalizable to perform adequately. As a diagnostic aid the performance of the software must have adequate sensitivity and specificity; however, the performance is dependent on the prevalence of the condition (i.e., pre-diabetes) being tested. Because this software is intended for a general population, the software may need to operate in consideration of a wide variety of patients in the intended population.

A.2.2 Context of Device Use

Characterization Feature	Considerations for Medical Device Software Risk Characterization
Intended User	•Does the software enable **new/different users** to achieve the clinical task than those who would perform the task without the software? •Does the user have the **expertise**, or access to the expertise, necessary to understand the inputs and/or outputs of the software?
Intended Use Environment	•Is use of the medical device software providing a clinical task or services in an **environment that would not otherwise have such services** available (e.g., would otherwise require an expert present)? •Is the device intended to be used in an uncontrolled or unconventional setting? •Can external factors, both physical and digital, affect the use, input or output of the device? •Do the expected virtual conditions and computing environment require additional software controls and/ or impact the users' access to the software?
Timing Within Healthcare Task/Intervention	•Does the user have **adequate time to review the basis** for the information output by the software or to review and curate the information being used as input to the software? •Could the software output **initiate a healthcare intervention that would not otherwise be identified** by a particular user or in a particular setting (e.g., pre-screening information prompting a patient to speak to a doctor about a possible condition)? •Are there **possible harms or dangers related to the healthcare task/intervention that could occur immediate** to the software's outputs? •Are there **possible harms or dangers related to the healthcare task/intervention that could occur at a time distant** from the use of the software, but that are related to decision points impacted by the software outputs or behaviour?
Role Of Software Output Within the Healthcare Task/Intervention	•Does an erroneous output from the software at the intended point in the workflow **put the patient on a path toward subsequent harm**? •Is the **frequency of output** appropriate to its role and timing in the workflow (e.g., is there a potential for notification fatigue)? •Does the software create a **single point of failure** in the clinical task/intervention?

Continuing the example, we consider questions related to the Context of Device Use:

In considering the *Intended User,* we recognize this medical device software enables both new and different users (i.e., different Health Care Providers (HCPs)) to achieve the clinical task (i.e., to identify individuals with pre-diabetes) that would otherwise not be performed without the software. This medical device software can be used by different intended users (i.e., different primary care and/or specialty HCPs). The software is analyzing health related data in electronic health records that does not require the user (i.e., HCP) to have specialized training. This medical device software requires the user (i.e., HCP) to have the necessary expertise to understand the input (i.e., type of data in electronic health records that the software analyzes) and the output (i.e., pre-diabetes) produced by the software.

The *Intended Use Environment* for this medical device software includes providing services in a healthcare (i.e., clinical) environment and is not intended to function outside healthcare settings or in those settings where healthcare is not being delivered with access to an electronic health record (i.e., settings using paper-based records). External factors (i.e., those factors that can impact the function of the medical device software) such as physical (e.g., physical related factors) and digital (e.g., broadband, internet connectivity, access issues to different healthcare databases) factors, may have a minor or negligible effect on the use, input, or output of the device. Further, the restricted intended use environment reduces the variability of operating conditions where the software must perform adequately.

As for the *Timing within Healthcare Task/ Intervention*, we recognize that the output of this medical device software is considered routine and non-urgent. The user has adequate time to review the output of this medical device software and to curate and review the basis or information used as its input. Because of the intended timing, the impact of the software's risks may overall be considered lower than those risks might be in a time-critical or urgent use case. Because some patients for whom review might be impactful to their future care could be missed if the software does not present their cases for review, there is a possible harm that could occur distantly from the use of the software.

When considering the *Role of Software Output within the Healthcare Task/Intervention*, we recognize that as a recommendation for further testing, the risk of output from the software at the intended point in the workflow putting the patient on a path toward subsequent harm is low. The frequency of output from the software and timing in the clinical workflow do not present risks of notification fatigue. The software also does not present a single point of failure in the clinical task/intervention as other data within the patient's primary care routine to identify symptoms of prediabetes.

As we consider questions related to the Context of Device Use, the *Intended User* for the medical device software in the scenario provided is limited to healthcare practitioners in the *Intended Use Environment* of a health care facility. This, in combination with the *Timing within Healthcare Task/Intervention* and *Role of Software Output within the Healthcare Task/Intervention* considerations indicates that these characterisation features pose a lower impact on overall risk characterization.

A.2.3 Device Function Use

Characterization Feature	Considerations for Medical Device Software Risk Characterization
Output Type	•Is the output **supplementing** additional information to contribute to a clinical interpretation or workflow recommendation? Is it a **replacement or substitution** for information meant to determine a clinical interpretation, workflow recommendation, or as data for use in a medical purpose? •Is the output commonly accepted in **clinical practice** or based upon sound **scientific principles**? Is the output **proprietary**? •Is access to the output **tiered or limited** by user or other credentials? •Is the output **Boolean**, e.g., values that are either true, or false?
Input Source	•Is the input source from a human user, medical device, or consumer product? •Is the input source **unique** or could the data be obtained through **other methods or sources**?

	•Is an adequate input source **governed by specific parameters** such as rate, sensitivity, or precision? •Is the input data direct or **informed or transformed** by other tools, products, or intermediaries? •Are there **multiple input sources** or data types? Are they **interdependent**?
Level of Task Automation	•Does the user **control generation** of the output? •Does the output require **review/approval** and allow **modification** by the user? •Are some outputs are flagged for review or provide a way to go back and **edit the output**, for example if assigned low confidence/high risk? •Can the user **review the basis** for the output? •Does the user **control or review** the software inputs?
Degree of Clinical Autonomy	•Is a user in the loop? Is the user in the loop a health care professional?
Intelligibility/Transparency/Explainability of Underlying Logic including the Algorithm/Technology used and How and Output is Reached	•Is the functionality of the product **explained and understood** by the user? •Is the functionality of the product **explained and understood** by users other than the patient? Is **different information provided** to different user groups or patients? •Is the functionality **partially explained** or **partially able to be evaluated** by the user? (e.g., output provided with saliency maps)**?**
Destination/Target of Output	•Is the output **the only instruction/data/information** needed to drive the target's next action?

Continuing the example, we consider questions related to the Device Function/ Use:

In considering the *Output Type,* we recognize this medical device software provides additional information (i.e., diagnosis of a pre-diabetes state) that supplements clinical recommendations (e.g., for subsequent diagnostic testing) with data that is used for a medical purpose (e.g., recommendations for lifestyle modification and/or treatments). The output of this medical device software is commonly accepted in clinical practice (i.e., the diagnosis of pre-diabetes) and, provided it has been adequately validated with an appropriate indication for use, is based on sound scientific principles. This medical device software output is considered proprietary, because the specific calculation to arrive at a threshold to present the output to the HCP for review is devised by the company and is not simply a well-known and accepted threshold or calculation. Access to the output of this medical device software is first made available to the HCP who ordered the use of this software (i.e., analysis of health-related data for an individual who does not have pre-diabetes to determine if pre-diabetes is present in this individual). Thereafter, the output of this medical device software is accessible by HCPs who are providing care to this individual and information is not withheld from the HCP on the basis of a specific product access tier. The information is also not meant to be shared with a wide variety of users such that varying levels of access is implemented, such as might be the case if the product's outputs were meant for review by both patients and their providers.

The *Input Source* of this medical device software is unique and limited to the data that is available in electronic health records for individuals in whom this software will be used. The input data cannot be obtained through other methods or sources. The input source of this medical device software is governed by specific parameters, notably structured data in electronic health records (e.g., diagnostic testing results, vitals measurements, demographic information). The input data of this medical device software is not transformed by other tools or products. This medical device software contains one input source (i.e., data in electronic health records) but includes multiple interdependent data elements (e.g., demographic data, laboratory and diagnostic testing results, treatments). These structured, regular data inputs from known sources of expected uniform quality do not appear to introduce novel or altered risks as a result of introducing the software solution. An HCP would review the same data to make an independent decision if the software was not available.

In considering the *Level of Task Automation,* we recognize the user does not control generation of the output for this medical device software. The output of this medical device software does not require review/approval or allow modification by the user, and the output of this medical device software does not enable retrospective editing of the output. Because of the nature of the task this software is meant to perform, which does not immediately impact a next clinical action without review and provides a new datapoint (threshold) rather than a modification of existing data, the level of task automation may not introduce risks specific for this device software. We also recognize the user can review the basis for the output of this medical device software and the user does not control or review the inputs for this medical device software. Only certain outputs of the software will be elevated to the HCP's attention, which suggests that the software solution may introduce a different risk than those present if the task was completed manually (e.g., failing to identify at-risk patients to present to the HCP that they might otherwise have noted if reviewing the data manually).

In terms of the *Degree of Clinical Autonomy*, a clinician is in the loop to review any outputs flagged by the software and to make the next decision in the clinical workflow – such as follow up tests for the patient. However, as noted above, a clinician will not be informed of patients who have *not* met the threshold to be considered "at risk" by the software.

Considering the *Intelligibility/Transparency/Explainability of Underlying Logic*, we recognize the functionality of this medical device software is explained (i.e., within its indication for use and ordering requirements) and is understood by the user. The functionality of this medical device software is explained to and can be evaluated by the user (i.e., input data includes structured data elements in electronic health records). The analysis (i.e., statistical or computational approach) is partially explained to the user.

Considering the *Destination/Target of Output,* this software likely does not provide an output that would be the sole instruction/data/information to drive the HCP user's next step. The output will present cases for the HCP to review and introduce a single new datapoint (patient has been identified as above a threshold). The HCP will have the patient's data for review in addition to the information that the patient has exceeded the threshold to help inform their next decision. However, as noted above, the HCP will not be presented with any data on patients who do not exceed the software's threshold, which could result in *no* decision made for such patients.

After consideration of questions related to Device Function Use it may be considered that *the output type*, supplementing additional information to contribute to a clinical interpretation or workflow recommendation, in this case a prediction or diagnosis commonly accepted in clinical practice or based upon sound scientific principles, may not greatly impact the risk of the device. However, the specific threshold calculation is proprietary and the software is replacing a manual review of a patient record and introduces the possibility of incorrectly filtering patients for review by the HCP.

A.2.4 Device Change Management

Characterization Feature	Considerations for Medical Device Software Risk Characterization
Degree of Learning/Change Management Autonomy	•Does the medical device software **independently change** its underlying algorithms? •**How often** is medical device software performance verified? •Are updates to algorithmic performance **driven by** non-clinical or clinical users, or manufacturer driven, or a combination of these users?
Domain of Learning/Change Implementation	• Is **domain specific implementation** relevant to the performance software? • Where are changes intended to be implemented and **how variable are the domains**?
Installation, Update and Error Correction Infrastructure	•What specific **channels** are used to **distribute** the medical device software? •Does the medical device software have **multiple installation locations**? Where are corrections initiated?

Last, we consider questions related to the Device Change Management:

In considering the *Degree of learning/change management autonomy*, we recognize this medical device software does not independently change its underlying algorithms. The performance of this medical device software is verified on an annual schedule by the product developers and validated by clinical users within the specific healthcare site. Updates to the algorithmic performance are monitored by clinical users and the manufacturer.

In *Domain of learning/change implementation*, we note that learning and/or change management may result in different accuracy or precision when this software is used across different clinical sites or regional locations (i.e., based on the demographic characteristics of the individuals in whom this software is used).

Regarding *Installation, Update and Error Correction Infrastructure*, we note that this medical device software's distribution channel is a web application, and that the software installation occurs on a server at the individual clinical site by clinical users.

In summary, for such a product, overall impact on risk posed, or introduced, by the software takes into consideration multiple characterization features across information groupings, and those that are most relevant to the particular device software may be different depending on the device's intended use/purpose. For this reason, it is critical to have a clear description of the software to help build an understanding of the role of the medical device software and its unique implementation. For this example device, the particular software solution may introduce risks related to the automation of a previously manual step and new failure points in the intended workflow. However, because of the device's medical purpose and context of use, these potential risks may not have notably high impact. These considerations can be taken together when considering how the decision to design this software solution may impact the overall risk of the device or raise different hazards.

Appendix E: Examples Comparing Specific Risk Considerations

As with Example A above, addressing each of the characterization features through corresponding questions is helpful for evaluating risk. The below questions are listed by information grouping to support comprehensive discussion of risk considerations.

The pairs of comparative examples below further illustrate the hazards to be extracted in risk analysis can differ based on the unique characteristics of a given medical device software.

Example 1: Software intended to provide a therapeutic experience to reduce and relieve pain.

Scenario 1.1: The software is intended to be used in conjunction with prescribed pain management medications to reduce and relieve pain in cancer patients undergoing chemotherapy.

Scenario 1.2: The software is intended to be used to reduce and relieve pain in osteoarthritis patients that cannot take other pain relief medication.

In both scenarios in example 1 above, the intended use of the medical device software is to provide therapy to reduce and relieve pain, where the cause of such pain (i.e., the *Intended Condition/Disease/Disorder and Grade/Stage/Level*) is not the primary distinguishing feature that contributes to understanding the risk of the medical device software. Rather, in this case, understanding whether the medical device software is intended to be used adjunctively (i.e., the *Medical Purpose*) contributes significantly to potential hazards considered in the risk analysis of the software.

In scenario 1.2, the software is meant to provide therapy for patients who cannot utilize other pain relief therapy. Because the software is itself intended as therapy and cannot be used with, or adjunct to, additional treatment, the risk of the software could be considered higher in scenario1.2 than 1.1. The failure of the software output to provide efficacious therapy may be considered a single-point failure for achieving the intent of patient pain reduction or relief, and therefore the intended medical purpose may contribute to the hazards considered in risk analysis more than the software used in conjunction with other therapy, described in scenario 1.1.

In these two scenarios for a similar medical device software, we see that within the *Medical Problem and/or Objective* information grouping, characterization features contribute to the risk of the software differently. For such a product, the *Intended Condition/Disease/Disorder and Grade/Stage/Level* does not solely impact the risk posed by the software, but a more detailed understanding of the *Medical Purpose* contributes to a more complete understanding of the medical device software's risk.

Example 2: Software that aggregates data and highlights trends from a wearable monitor (DHT) for patients diagnosed with heart failure

Scenario 2.1: The software is intended to aggregate data and highlight trends from a wearable monitor for patients diagnosed with heart failure to help patients monitor their risk of hospitalization. The software helps to provide simple data visualizations to better understand the patient's longitudinal data, such as tracking an individual's health, care usage, and outcomes over time.

Scenario 2.2: The software is intended to aggregate data and highlight trends from a wearable monitor for patients diagnosed with heart failure to help patients and their healthcare provider with longitudinal data about the patient's heart health. The software provides simple data visualizations, including highlighting trends, to help the healthcare provider monitor their patient's risk of hospitalization between regularly scheduled visits and could be used to inform treatment-related decisions.

In example 2 above, the intended user for the medical device software in scenario 2.1 is limited to patients seeking to obtain more information about their own condition. In scenario 2.2, healthcare providers are included in the intended user group and have access to the data in addition to the patient themselves. In this case, a health care professional has specialized training that provides them with additional context to understand the data and trends the medical device software is highlighting, which a patient may not have. For this reason, it might be considered that the *Intended User* in scenario 2.2 may reduce hazards considered in risk analysis more than scenario 2.1, because at least one intended user in scenario 2.2 has expertise and training to appropriately understand and respond to the data they are receiving. The health care professional is provided access to the data such that it is not essential for the patient to independently identify if and when their data should be conveyed to their doctor.

However, it may also be worth considering that there is greater variability in the *Intended Users* of the medical device software in scenario 2.2 than scenario 2.1, because of the introduction of the clinician user. This difference also impacts the understanding of risk posed by the software, where the information must be conveyed adequately and appropriately to the different user groups. It is important to consider that multiple factors may influence the risk associated with any given characterization feature – a clinician or trained user does not always independently indicate a decrease or increase applicable hazards in the risk analysis of a device.

Example 3: Software function that uses physiological data captured on a wearable consumer product to determine the severity of symptoms in a patient with Parkinson's disease.

Scenario* 3.1*: The software is intended to aggregate measurements obtained from a regulated medical device and analyzed to monitor the severity of symptoms such as tremor in a patient with Parkinson's disease.

Scenario 3.2: The software is intended to aggregate measurements obtained from a wearable consumer product and analyzed to monitor the severity of symptoms such as tremor in a patient with Parkinson's disease.

In example 3 above, the *Input Source* in scenario 3.1 is limited to measurements obtained by a regulated medical device. In scenario 3.2, measurements are obtained by a wearable consumer product that is not subject to regulatory oversight as a medical device. In this case, the wearable consumer product may allow for expanded opportunities for collecting patient data, however the aspects of the performance of the wearable consumer product may be outside of the control of the developer. For this reason, it may be considered that the *Input Source* in scenario 3.2 may pose more applicable hazards for risk analysis than in scenario 3.1, because the manufacturer developing the software may not have life cycle control over the source of the data it is analyzing to monitor the severity of symptoms. In this case, additional steps may be necessary for the manufacturer to monitor performance of the wearable consumer product and to communicate any changes in performance to the user. In contrast, scenario 3.1 which obtains measurements form a regulated medical device, benefits from the verification and validation needed to obtain authorization (in cases where the intended use is fit for purpose), which may reduce applicable hazards due to a greater accuracy and precision of measurements of a product developed for the intended use. Regulations applicable to software using consumer products to perform regulated device functions vary by jurisdiction.

IMDRF International Medical Device Regulators Forum

906

Please visit our website for more details.

www.imdrf.org