ITUPublications

International Telecommunication Union
Standardization Sector

ITU-T Focus Group Deliverable

(03/2023)

Focus Group on Artificial Intelligence for Health (FG-AI4H)

FG-AI4H DEL04

AI software life cycle specification

ITU-T FG-AI4H Deliverable DEL04

AI software life cycle specification

Summary

Building a quality product includes performing quality tasks throughout the development lifecycle. For example, having a plan that describes the process that you use to develop software is better than not having a plan. Having product requirements in a documented form is better than having product requirements only in people's minds. Documenting the design of the software at both a high-level architecture as well as a unit-level helps tremendously when trying to maintain a piece of software that was written years ago, and the original developers are no longer available to help.

Due to the "black box" nature of some machine learning (ML) applications, having quality processes in place will be a significant factor affecting product quality and performance.

DEL4 provides an overview of existing software development lifecycle standards and how they can be applied to the development of health artificial intelligence (AI) applications.

It should be noted that risk management is mentioned in this Technical Report as well as in FG-AI4H-P-1 Risk Governance in Artificial Intelligence for Health [FG-AI4H-P1].

Keywords

AI for health (AI4H), software life cycle.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

Change Log

This document contains Version 1 of the Deliverable DEL04 on "*AI software life cycle specification*" approved on 24 March 2023 at the ITU-T Focus Group on AI for Health (FG-AI4H) meeting held in Cambridge, MA, USA, 21-24 March 2023.

Editor: Pat Baird
Philips
USA
Tel: +1 262 239 9321
Email: pat.baird@philips.com

Table of Contents

ITU-T FG-AI4H Deliverable DEL04

AI software life cycle specification

1 Scope

This deliverable includes the following considerations:

a) Identification of standards and best practices that are relevant for the AI for health software life cycle. Similar to other software life cycle processes, the AI software life cycle process needs to be specified;

b) Summary and critical review of the identified documents including a discussion of their limits/gaps and need for action;

c) Identification of a data development plan that supports life cycle steps that are specific / characteristic for AI for health software, such as training and test procedures based on data that potentially need to be annotated for safety and security risk management activities;

d) Specification of the AI for health software life cycle and definition of best practices for the different life cycle steps in one document (under consideration of a, b, and c). Overview and examples of best practices.

2 References

[FG-AI4H-P1] FG-AI4H-P-1 *Risk Governance in Artificial Intelligence for Health.* <https://www.itu.int/en/ITU-T/focusgroups/ai4h/Pages/default.aspx>

[IEC 62304] IEC 62304:2006, *Medical device software - Software lifecycle processes*. <https://webstore.iec.ch/publication/6792>

[IEC 82304-1] IEC 82304-1:2016, *Health software – Part 1: General requirements for product safety*. <https://www.iso.org/standard/59543.html>

[ISO 14971] ISO 14971:2019, *Medical devices — Application of risk management to medical devices*. <https://www.iso.org/standard/72704.html>

[ISO/IEC JTC 1/SC 42] ISO/IEC JTC 1/SC 42 *Artificial intelligence*. <https://www.iso.org/committee/6794475.html>

[AAMI TIR57] AAMI TIR57:2016, *Principles For Medical Device Security – Risk Management.* <https://webstore.ansi.org/standards/aami/aamitir572016>

[BS/AAMI 34971] BS/AAMI 34971:2023, *Application for ISO 14971 to machine learning in artificial intelligence. Guide.* <https://standardsdevelopment.bsigroup.com/projects/2020-02770#/section>

[CTA-2107] Consumer Technology Association CTA-2107 (2022), *The Use of Artificial Intelligence in Health Care: Managing, Characterizing, and Safeguarding Data.* <https://standards.cta.tech/apps/group_public/project/details.php?project_id=675>

[DIN 92001-1] DIN SPEC 92001-1:2019-04, *Artificial Intelligence – Life Cycle Processes and Quality Requirements – Part 1: Quality Meta Model*. <https://www.beuth.de/en/technical-rule/din-spec-92001-1/303650673>

[DIN SPEC 92001-2] DIN SPEC 92001-2, *Artificial Intelligence – Life Cycle Processes and Quality Requirements – Part 2: Robustness*. <https://www.din.de/resource/blob/792080/5cc278f717df1da1eabbd27fa5f13e2d/case-study-din-spec-92001-2-data.pdf>

[IMDRF] IMDRF (2018), *Essential Principles of Safety and Performance of Medical Devices and IVD Medical Devices*. Good Regulatory Review Practices Group. <http://www.imdrf.org/docs/imdrf/final/technical/imdrf-tech-181031-grrp-essential-principles-n47.pdf>

[Lomonaco] Vincenzo Lomonaco (2017), *Why Continual Learning is the key towards Machine Intelligence*. <https://medium.com/continual-ai/why-continuous-learning-is-the-key-towards-machine-intelligence-1851cb57c308>

[NIST Cyber 1.1] NIST (2018), *Framework for Improving Critical Infrastructure Cybersecurity Version 1.1*. <https://doi.org/10.6028/NIST.CSWP.04162018>

[Urbanowicz] Ryan Urbanowicz (2018), *An Introduction to Machine Learning*. <https://ldi.upenn.edu/wp-content/uploads/archive/Introduction-to-Machine-Learning.pdf>

[Xavier] Xavier Health, (2018), *Perspectives and Best Practices for Artificial Intelligence and Continuously Learning Systems in Healthcare*. <https://www.exhibit.xavier.edu/cgi/viewcontent.cgi?article=1024&context=health_services_administration_faculty>

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Technical Report

None.

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

IMDRF	International Medical Device Regulators Forum
ML	Machine Learning
OTS	Off-The-shelf Software
PAS	Publicly Available Standard
SaMD	Software as a Medical Device
SOUP	Software Of Unknown Provenance

TPLC Total Product Life Cycle

5 Conventions

None.

6 Resource landscape

There are several health software and AI lifecycle standards and guidance documents that have been published. This report uses the advice and experience provided in those documents. A description of this landscape is given below. Note that there are additional projects being started all the time – for example, [ISO/IEC JTC 1/SC 42] has a project proposal to develop an AI lifecycle process for all industries – not just healthcare. Therefore, a periodic review of current lifecycle standards might result in additional insights and updates to this Technical Report.

6.1 DIN SPEC 92001-1, Artificial intelligence – Life cycle processes and quality requirements – Part 1: Quality meta model

[DIN 92001-1] is a publicly available standard (PAS) freely available from the DIN public website. This paper is intended for multiple industries (not just healthcare) and provides a high-level framework for a quality metamodel, as shown in Figure 1.

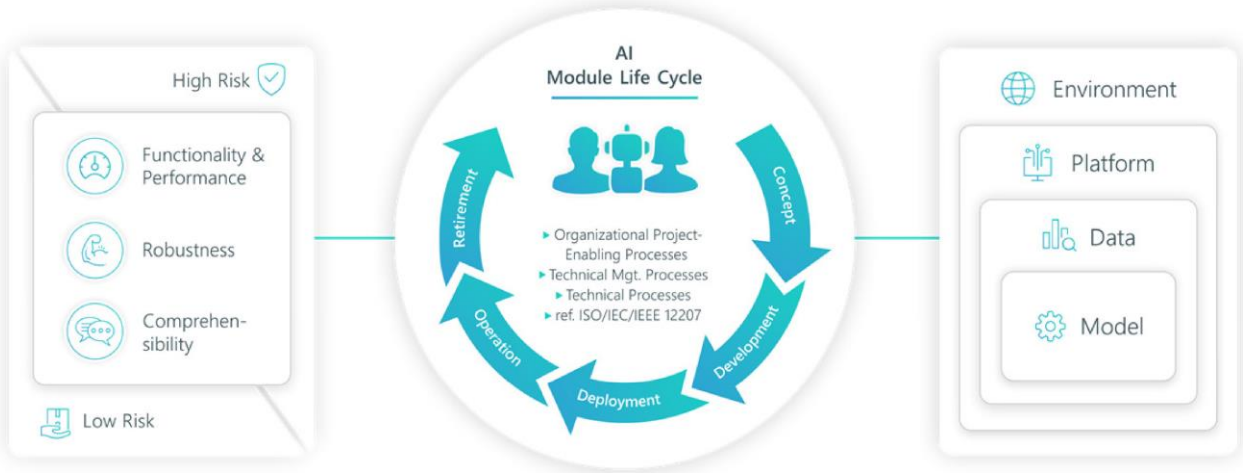

Figure 1 – DIN SPEC 92001-1 high-level framework for a quality metamodel

6.2 DIN SPEC 92001-2, Artificial intelligence – Life cycle processes and quality requirements – Part 2: Robustness. [DIN SPEC 92001-2]

6.3 IEC 62304:2006 Medical device software: software lifecycle process [IEC 62304]

This is a commonly used standard for medical device software. Note that the current version of the standard was published in 2006, and was intended to apply to the embedded software, as the concept of "Software as a medical device (SaMD)" had not really been considered at that time. This means that the IEC 62304 standard assumes that there are lifecycle processes in place which are applicable to the entire product (and not just software) that are covered in other standards. For example, establishing up-front user needs and validating that the needs have been met are outside of the scope of [IEC 62304], as those process steps could involve hardware aspects of the device. With the rise of software as medical device (SaMD) applications, this gap was noticed and the [IEC 82304-1] software standard was developed to address the gap. This paper includes both sets of lifecycle activities.

It should be noted [IEC 62304] has updated and published its revised standard and this deliverable should be reviewed and updated if necessary.

Within the software development lifecycle, this Venn diagram shows the overlap between the different domains that are involved – there is a subset of AI that is machine learning, and there is a subset of statistics that is data science, and these items overlap in many ways. The scoping of data management and the process for AI development have different but overlapping processes to manage safety risk management, privacy risk management, and cybersecurity risk management.

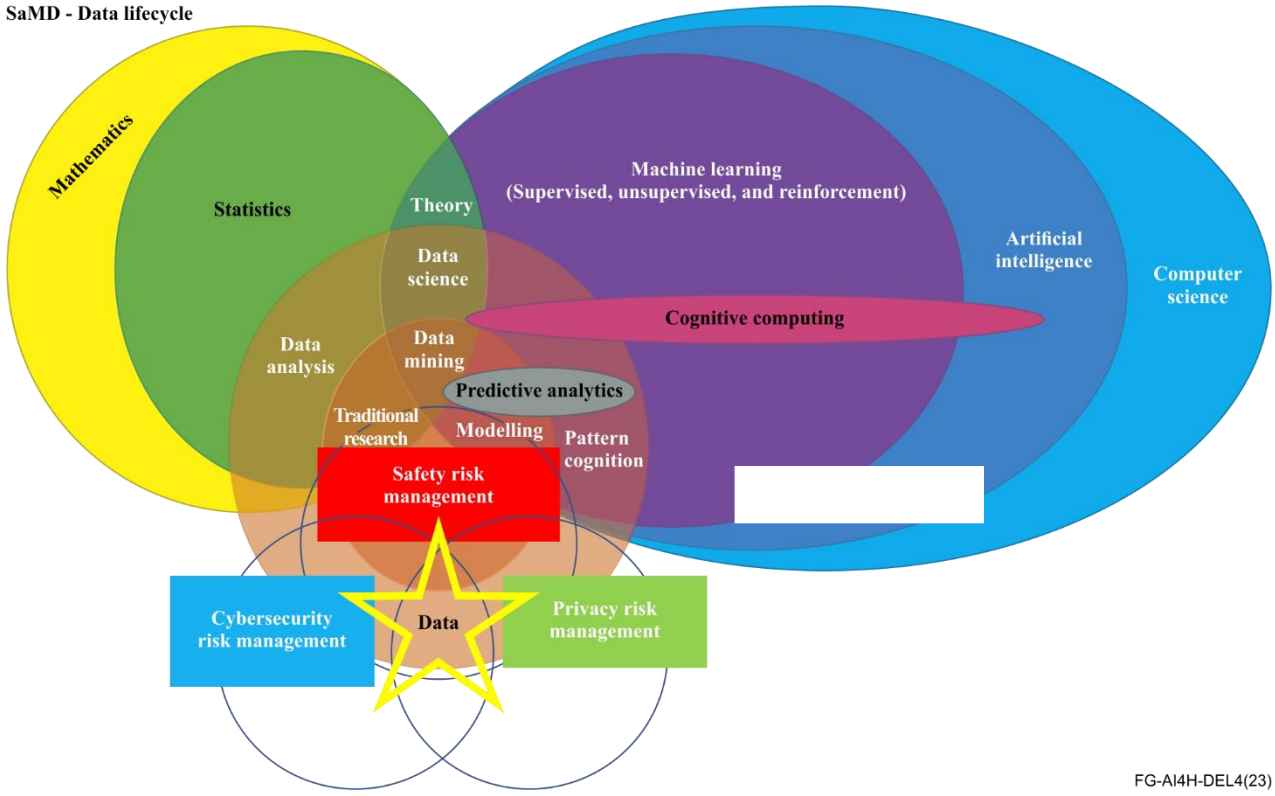

Figure 2 – AI/ML and SaMD: Modified from "An introduction to machine learning", Ryan Urbanowicz, Perelman School of Medicine, University of Pennsylvania. [Urbanowicz]

6.4 Machine learning data plan

It is recommended that a data plan be created for the machine learning lifecycle that supports the software and algorithm development. Data elements to inform the design and development for risk management, testing and monitoring including updates should be established.

If we take a look at the product roadmap with respect to AI/ML development and the total product life cycle (TPLC), essentially the main takeaways are to understand the various data quality elements that are contributing to the software algorithm starting from initiation to distribution. A data plan is recommended to describe the various activities required for data acquisition, pre-processing, training, evaluation and operations.

Figure 3 – Machine learning data plan

6.5 NIST cybersecurity framework 1.1

The software development lifecycle should include machine learning methods that have requirements for anomaly detection [NIST Cyber 1.1] by following a risk-based approach for potential security/privacy issues. Software system architecture and design should consider using supervised and unsupervised learning in the detection of anomalies.

Table 1: Function and Category Unique Identifiers

Function Unique Identifier	Function	Category Unique Identifier	Category
ID	Identify	ID.AM	Asset Management
		ID.BE	Business Environment
		ID.GV	Governance
		ID.RA	Risk Assessment
		ID.RM	Risk Management Strategy
		ID.SC	Supply Chain Risk Management
PR	Protect	PR.AC	Identity Management and Access Control
		PR.AT	Awareness and Training
		PR.DS	Data Security
		PR.IP	Information Protection Processes and Procedures
		PR.MA	Maintenance
		PR.PT	Protective Technology
DE	Detect	DE.AE	Anomalies and Events
		DE.CM	Security Continuous Monitoring
		DE.DP	Detection Processes
RS	Respond	RS.RP	Response Planning
		RS.CO	Communications
		RS.AN	Analysis
		RS.MI	Mitigation
		RS.IM	Improvements
RC	Recover	RC.RP	Recovery Planning
		RC.IM	Improvements
		RC.CO	Communications

Figure 4 – NIST cybersecurity framework 1.1

6.6 IEC 82304-1:2016, Health software – Part 1: General requirements for product safety. [IEC 82304-1]

As mentioned previously, [IEC 82304-1] was developed to fill the lifecycle gap from IEC 62304. Figure 5 shows the additional process steps that are included in [IEC 82304-1]. It can be noticed that

the extra process steps occur at the very beginning and very end of the product development cycle, and extends into post-market activities.

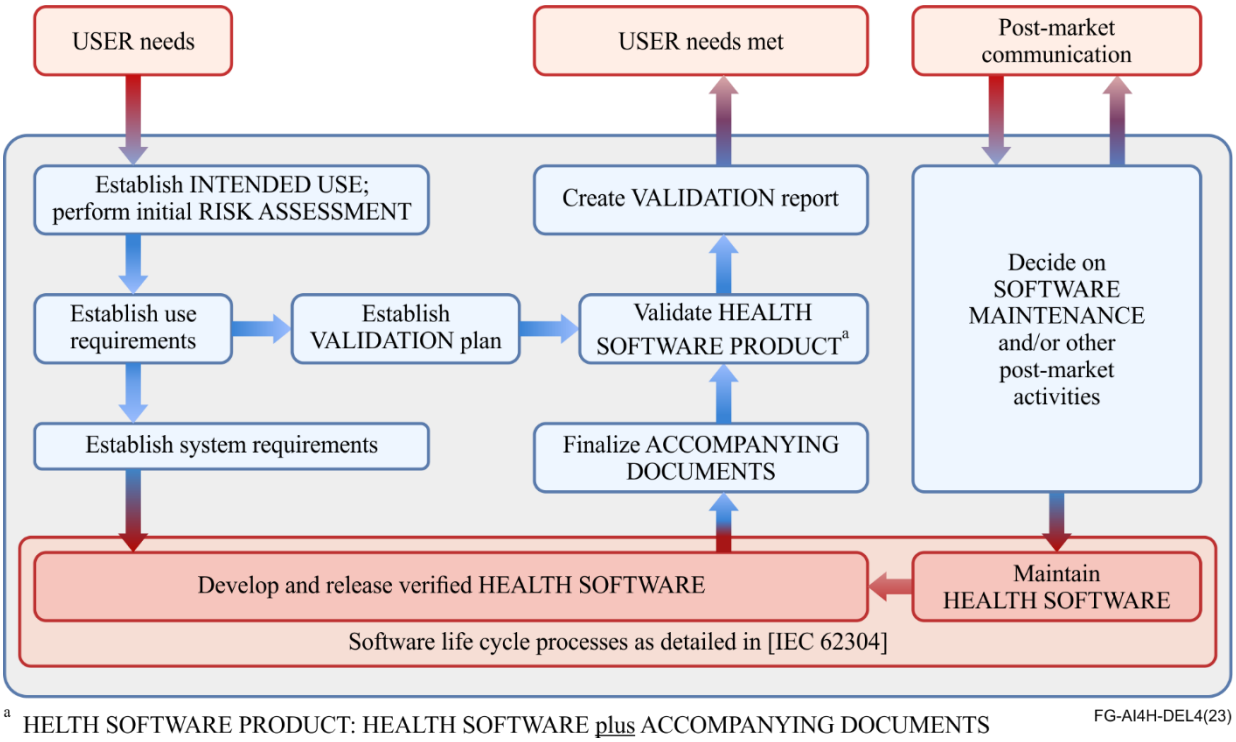

Figure 5 – Additional process steps included in [IEC 82304-1]

6.7 Product security requirements and framework

Security and privacy requirements should be taken into consideration as part of a secure design for a defense-in-depth strategy and security risk management. The product security framework encompasses the premarket SaMD design controls (blue pillars), and the post market support and services requirements, which should also be considered as part of the design and development lifecycle.

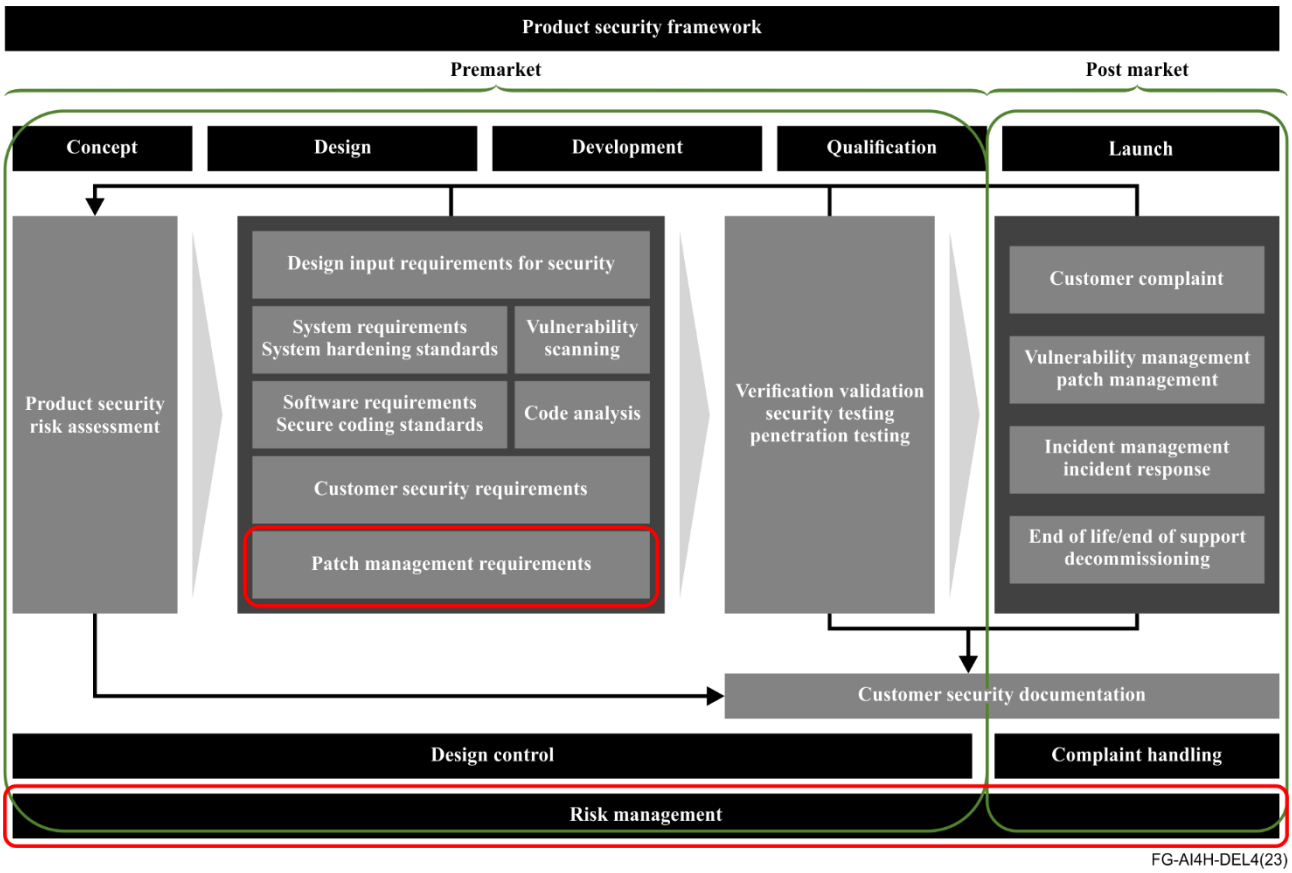

Figure 6 – JSP product security framework

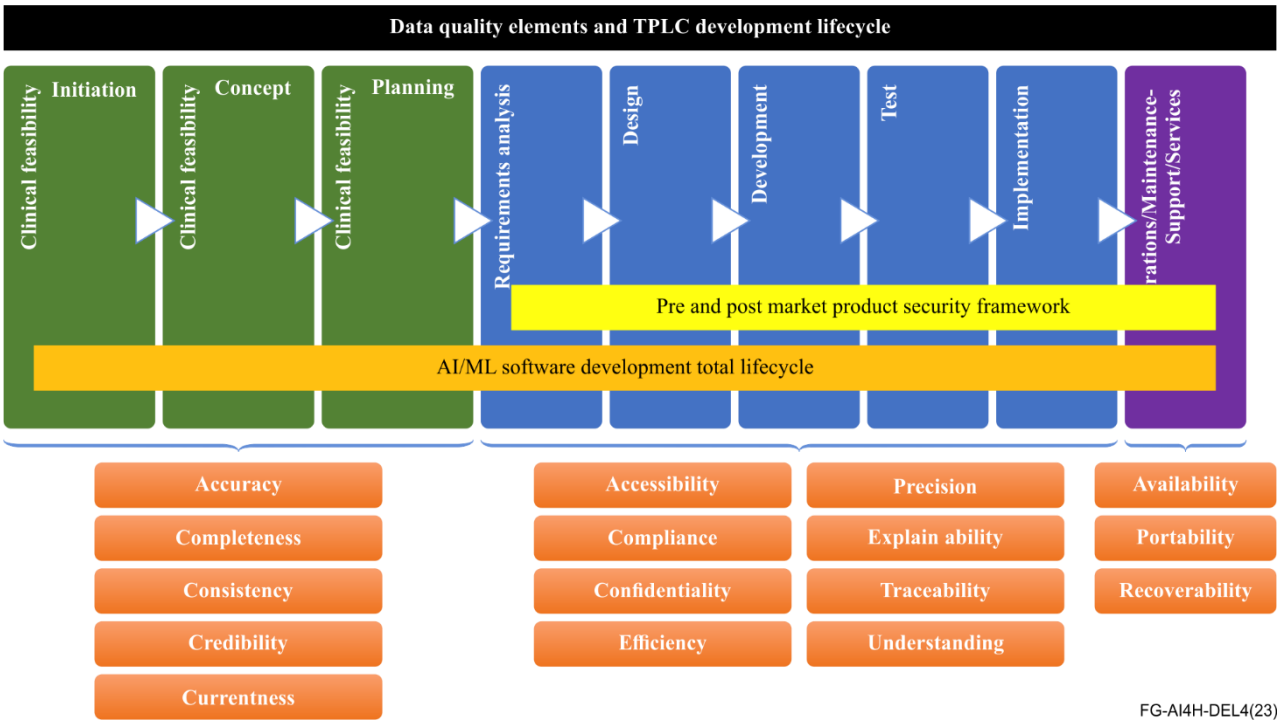

Figure 7 – Total product life cycle (TPLC) and product security framework

6.8 Perspectives and best practices for AI and continuously learning systems in healthcare, Xavier Health, [Xavier]

Anticipating that there would eventually be a need for an AI lifecycle process, a group of volunteers from the Xavier Health initiative started developing a paper to address some of the gaps between the

[IEC 62304] and [IEC 82304-1] standards and the needs of machine learning systems. This whitepaper was published in 2018 and was used as the primary basis for the development of this report.

6.9 IMDRF (2018), Essential principles of safety and performance of medical devices and IVD medical devices. Authoring group: Good regulatory review practices group. [IMDRF]

This guidance document from the IMDRF contains a significant amount of advice regarding aspects of a device that support a quality product. Some principles apply to AI systems, some principles do not, and some principles can be adapted to address the unique needs of AI. A series of recommended updates to the essential principles document were in the process of being developed and were submitted to the IMDRF by the end of 2020.

7 Discussion of residual gaps for AI

Traditional software lifecycle guidance documents are written for traditional software applications, and much of what they suggest is applicable to a wide range of AI applications. However, additional lifecycle requirements might be needed for AI and ML applications, due to any unique aspects of those applications.

For example, learning systems have additional process steps that should be planned and accounted for. Some of these steps are in the diagram of Figure 8 (adapted from [Lomonaco]).

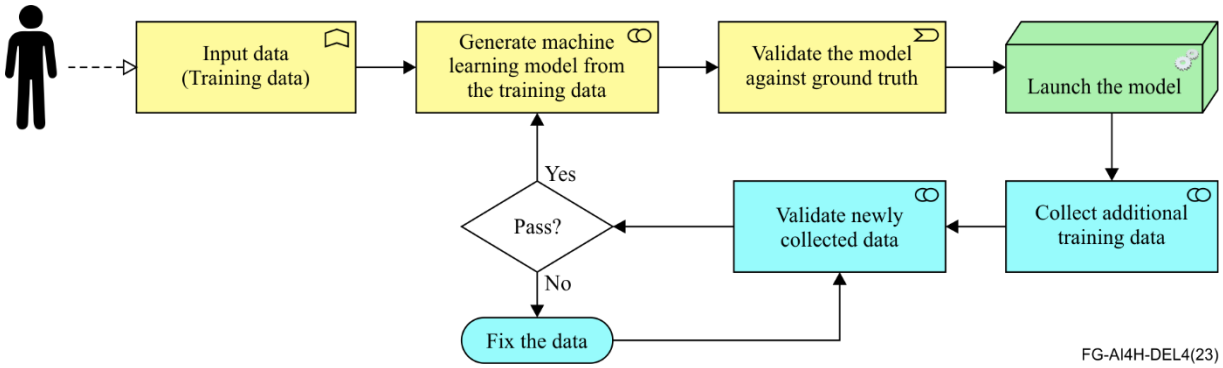

Figure 8 – Additional process steps needing accounting and planning

A review was performed of the papers listed from clause 6 onwards, and a high-level summary of the resulting lifecycle requirements (including closing gaps) is listed in clause 6 as well.

It should be noted that [IEC 62304] and [IEC 82304-1] are widely used for both regulated and unregulated medical devices and health software; there is currently a project inside their parent standards committee to assess what AI-related gaps exist in those and other related standards. Specific recommendations are being developed for updating [IEC 62304] and [IEC 82304-1].

7.1 Consumer technology association. The use of artificial intelligence in health care: managing, characterizing, and safeguarding data, 2022 [CTA-2107]

This free standard discusses the data lifecycle for ML applications in healthcare and defines the activities for the phases listed in the data lifecycle diagram.

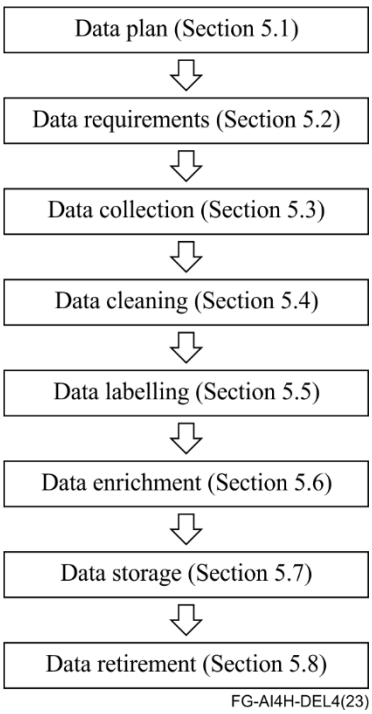

Figure 9 – Data lifecycle from [CTA-2107]

7.2 Proposed solution

The following lifecycle is proposed, based on the standards cited above. Note that some of these process steps can occur in parallel and a project team may decide to develop these deliverables incrementally – for example, they may choose to identify product features that are of high value to the user and develop requirements, design, implementation, and testing for that subset first, and once that is completed, begin working on the next highest-value features. Stated differently, this lifecycle approach does not imply that a "waterfall" approach is needed; alternative development techniques (such as agile software development) are acceptable.

7.3 Establish intended use ([IEC 82304-1])

The intended use should be documented, including a description of the intended user and the use environment for the product.

7.4 Perform initial risk assessment ([IEC 82304-1])

Based on the intended use, user, and environment, an initial identification of potential risks shall be performed and documented.

7.5 Establish use requirements ([IEC 82304-1])

High level user requirements shall be documented. These can be written from a user's point of view. Note that these are different than "usability" requirements which focus on ergonomics, human factors, etc. "Use" requirements are a high-level description of what the product needs to do to satisfy the user needs and the intended use.

7.6 Establish system requirements ([IEC 82304-1])

Requirements for the system's performance and functionality should be documented. This may include requirements related to the hardware platform, interoperability, and high-level functionality.

7.7 Create software plan(s) ([IEC 62304], [Xavier])

There should be a plan that outlines the planned software development activities that should complement the machining learning data plan as mentioned in clause 6.4. This includes change management and version control. For learning systems, this plan would also include process steps to ensure training and test data integrity, reliability and validity.

7.8 Software requirements ([IEC 62304], [Xavier])

Requirements relating to the software aspects of the product shall be documented, including details of interfaces, alarms, security, usability, and data requirements.

7.9 Software architecture ([IEC 62304], [Xavier])

The architecture of the system shall be documented. For example, what are the major systems and subsystems of the product? How do those systems relate to each other? Are there external interfaces with other products that affect the design (e.g., connection to other equipment, cloud storage, etc.)? How will data be warehoused? What data extraction or processing tools are used?

7.10 Software detailed design ([IEC 62304], [Xavier])

The details of the software design should be documented (e.g., a unit-level description of the software.) This includes internal interfaces between different parts of the software and external interfaces.

7.11 Software integration, unit-level, including testing ([IEC 62304], [Xavier])

The unit-level software code shall be created and shall be verified at the unit level. The verification activities should be documented.

7.12 Software integration, integration level, including testing ([IEC 62304], [Xavier])

Software components and subsystems will need to be integrated and verified. The verification activities should be documented.

7.13 Software system testing ([IEC 62304], [Xavier])

The software should be tested at a system level, including external interfaces. The verification activities should be documented.

7.14 Software release ([IEC 62304], [Xavier])

Before the product is used for validation activities and before the product is released, the developer should ensure that all of the process steps have been followed for the development and testing of the product. Any known defects should have an impact analysis performed and documented, as well as a justification for releasing the product with these defects. To help support future improvements and root cause analysis, the software and associated documentation should be archived.

7.15 Establish validation plan ([IEC 82304-1])

Although a product may meet all of its documented requirements, it does not mean that the product actually meets the needs of the users. "Validation" in the medical device industry is a process where you determine if the product meets its intended use and satisfies user needs. Note that this is a different definition of "validation" than what is typically used in data science.

A validation plan should include a description of how the product will be validated, including requirements on the type of users to be recruited, use environment, tasks to be performed, test methods (including simulated use, if needed), and success criteria.

7.16 Validate the product ([IEC 82304-1])

The software should be validated, and the validation results documented. This includes any deviations from the expected performance. If any design changes are needed, the reason for the change will be documented, and an assessment of the need to re-verify or re-validate will be performed.

7.17 Create validation report ([IEC 82304-1])

The report includes the validation results, and analysis of the results, and a statement regarding the acceptability of the product risk.

7.18 Monitor product performance ([IEC 82304-1], [Xavier])

After the product is launched, product performance should be monitored, even on non-learning systems. Customers often have feedback about the product, including ideas for improvements, finding software defects, additional user needs, etc.

Therefore, there needs to be a mechanism for people to leave feedback on the product (e.g., telephone, email address, website, etc.) and a process for that feedback to be analysed and a decision made on whether or not the application needs to be updated.

In the healthcare industry there are often papers written about a particular topic or a study that is performed, papers about competitive products, product recall notices about similar products, changes in clinical practices over time, etc. and these publications are a good source of feedback for the developer.

These potential data sources should be identified and monitored. The monitoring activity can happen at different rates, for example, monitoring customer feedback might occur on a daily basis, but a journal review might only be necessary every few months.

The monitoring process shall be documented and if a change is needed, it would follow the change control and configuration management process.

7.19 Maintain software after launch ([IEC 82304-1])

Often changes need to be made to software after the initial product launch. Some of these changes may be to correct defects in the software, other changes may be simple enhancements to the product, and other changes may have a significant impact on product behaviour or performance.

For example, a customer might have a complaint about the software and the root cause of the complaint could be an incorrect requirement, a design flaw or it could simply be a coding error. Or the vendor that supplies the operating system has released a cybersecurity patch and this update needs to be sent to the customers quickly. Regardless of the source, an impact assessment should be performed (including any impact to risk), the appropriate documentation should be updated, the software change should be verified, and an analysis performed if the product needs to be re-validated.

It should be noted that changes to released software may require regulatory action however, that is out of the scope of this lifecycle report.

7.20 Retirement ([IEC 82304-1])

There comes a time when a product will no longer be supported by the developer. When it is time to retire a product, a plan should be in place regarding how the users will be notified and how privacy / security will be maintained (e.g., what to do with the central database that holds the patient information?).

7.21 Risk management (continuous) ([IEC 82304-1], [Xavier])

The product lifecycle should include the risk management processes such as safety risk management as described in [ISO 14971]. The key process steps are shown in Figure 10. Since ML systems have

new failure modes as compared to traditional software development, a Technical Report has been developed: [BS/AAMI 34971]:2023 Application of ISO 14971 to machine learning in artificial intelligence – Guide. In addition, security risk management should be considered as part of secure design and safety risk management activities [AAMI TIR57].

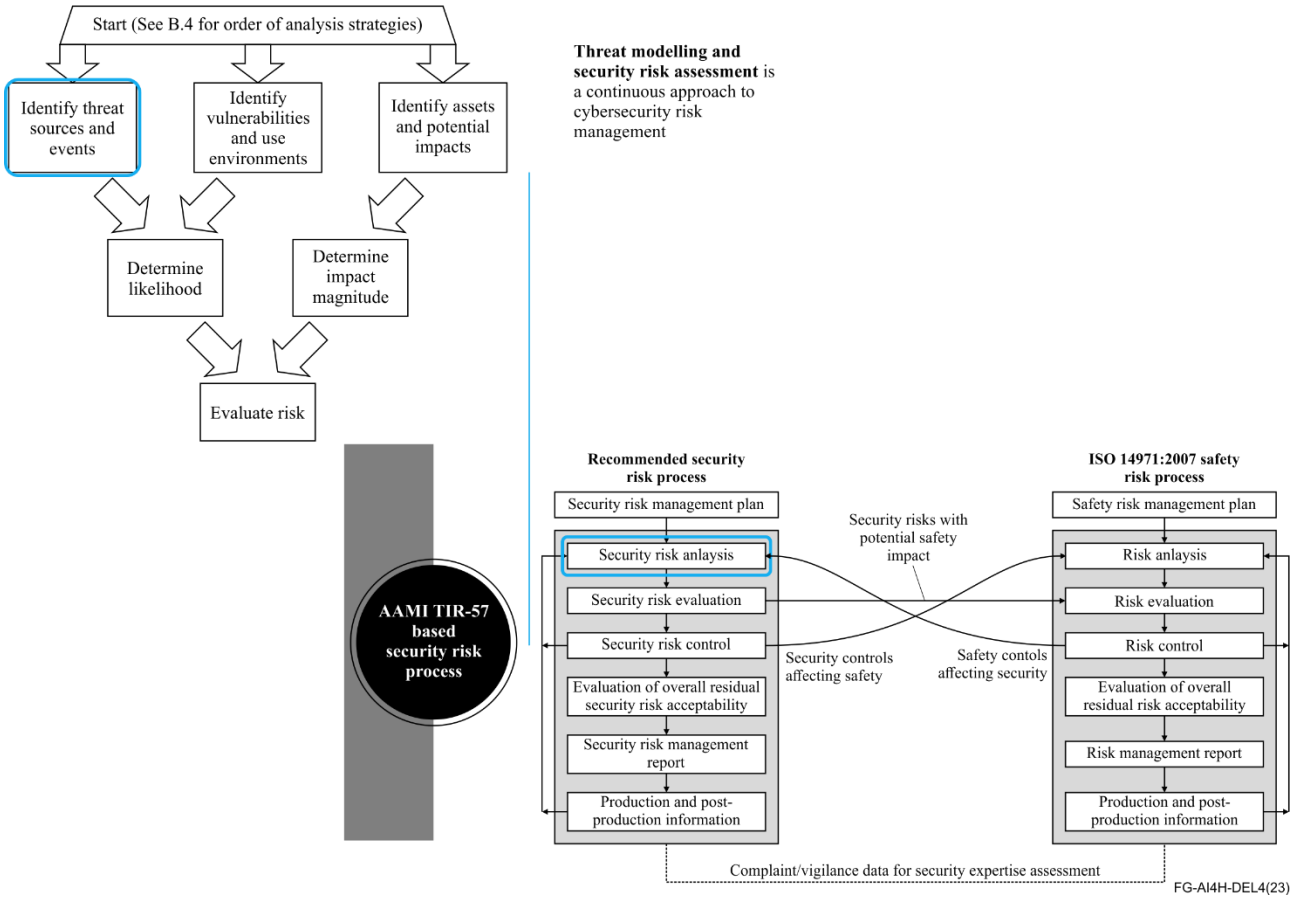

Figure 10 – Security risk and safety risk management interdependencies

7.22 Change control / problem resolution (continuous) ([IEC 62304], [Xavier])

Defects will occur. Defects may be due to poor or incorrect requirements, software implementation errors, errors in third party software (sometimes called "off the shelf (OTS)" or "software of unknown provenance (SOUP)"), or there may be ideas for new features or changes to existing features.

It is common to have a database or other tracking tool to manage change requests, and the impact that the change will have on the product files (e.g., is this just a code change or are new requirements needed? What re-testings are needed?), who is assigned to perform the change, and the current status of the change. Before being implemented, the changes should be approved by the appropriate level of management.

7.23 Configuration management (continuous) ([IEC 62304], [Xavier])

During development, the software is constantly changing. Even after the product is launched, there will likely be changes – new features, fixing defects and so on. Therefore, a plan needs to be in place that describes how the software and documentation will be version controlled.

After the product is launched, new versions could be created, and there may be multiple versions in use across the world. It is important to manage these different versions as there may be future upgrades, software patches, fixing defects, etc., and knowing what upgrades are needed to which version is vitally important. Even during development there may be instances where the developers

are working on a new iteration while usability testing is being performed on an older (but more stable) version of the software.

There should be a documented configuration management process including a description of how new versions will be released to the customers.

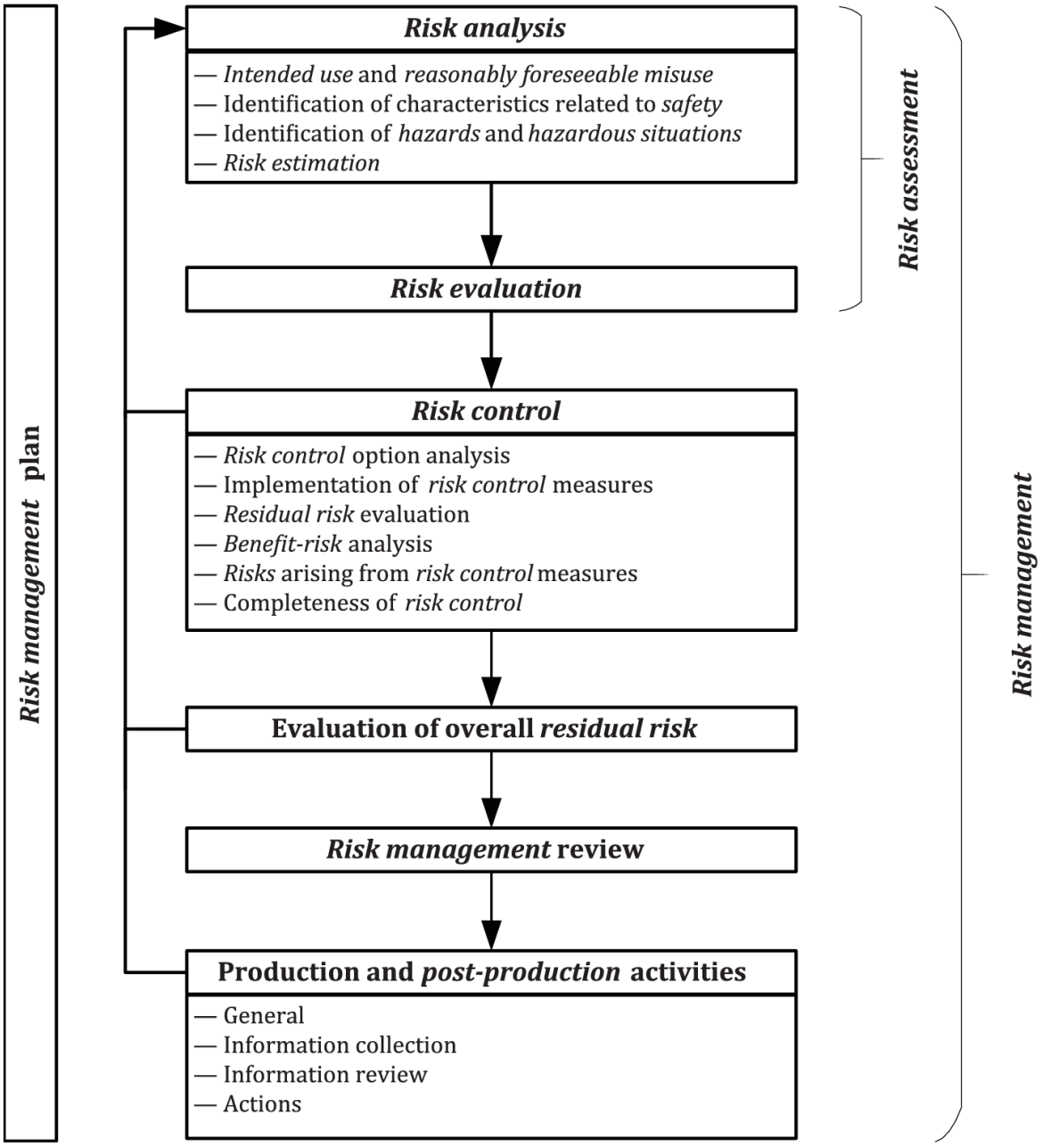

Figure 11 – Key risk management process steps (Source: [ISO 14971])