The new integrated model of medical software: IEC 62304 & IEC 81001-5-1

Medical software can be fully compliant with IEC 62304 and function properly, until it is compromised by a vulnerability exploited in a real-world context. This is the breaking point of the traditional model: safety compliance is no longer sufficient to ensure patient safety in a connected ecosystem.

Why can a system compliant with IEC 62304 still be unsafe in today's real-world environment?

In modern medical software, there is no longer a real separation between safety and cybersecurity: medical software fails not only because of software bugs, but also when it is attacked, with direct consequences on patient safety.

Connected devices, digital healthcare ecosystems and distributed software have fundamentally changed the way risk is managed: it is no longer enough to ensure that software is working properly, but that it is protected from external attacks.

It is in this context that the relationship between IEC 62304 and IEC 81001-5-1 becomes central: they are not two alternative standards, but complementary perspectives of the same problem: building reliable, secure and resilient medical software throughout its entire life cycle.

IEC 62304: The Regulatory Basis of Medical Software

IEC 62304 still represents the regulatory basis for the development of medical software and continues, twenty years after its publication, to be a consolidated international reference for software life cycle management in medical devices.

Its goal is to define a structured framework for the software life cycle, with a clear focus: to ensure that the behavior of the software does not introduce unacceptable risks for the patient.

The main concepts on which IEC 62304 pays the most attention are:

- the software life cycle: from development to maintenance;
- software risk classification: Class A, B, C;
- traceability between requirements, implementation and testing;
- risk management;
- the management of anomalies and changes.

In other words, IEC 62304 answers a specific question:

"Does the software work properly and securely with respect to its design?"

It is a model strongly oriented towards **safety**, i.e. the "freedom from unacceptable risks" [1], to prevent damage resulting from errors, defects or internal malfunctions.

IEC 81001-5-1: Cybersecurity in the Medical Software Lifecycle

IEC 81001-5-1 extends the traditional model by introducing the cybersecurity dimension **into the life cycle of medical software**.

It does not just define technical controls but integrates cybersecurity into the development and maintenance processes. An important aspect is that this standard stems from the experience gained in the world of industrial automation and control systems, transposing and adapting the principles of IEC 62443-4-1, an international reference for the Secure Development Lifecycle in industrial systems, to the context of medical software [2].

The main aspects reported by IEC 81001-5-1 are:

- Secure Development Lifecycle (SDL)
- Threat Modeling
- Security Requirements
- Ongoing vulnerability management
- Software Bill of Materials (SBOM)
- Patch & Update governance

These elements are not only good practices for safe development but are also playing an increasingly important role from a regulatory point of view. The application of IEC 81001-5-1 is now increasingly used as a reference to demonstrate compliance with the general security requirements of the European MDR and IVDR regulations, considered one of the main expressions of the state of the art for cybersecurity management in medical software. This approach is also consistent with the latest FDA guidelines on cybersecurity in medical devices.

The question it introduces is profoundly different from IEC 62304:

"Does the software remain secure even in a hostile, unpredictable, and ever-changing environment?"

Here the focus is no longer on the error, but on **the intentional attack** and the ability of the system to resist, detect and react [3].

From software lifecycle to secure software lifecycle

As the model shifts from "internal error" to "external threat," the nature of the risk also changes.

A malfunction is no longer the only source of danger: for example, infusion control software can fully comply with the requirements of IEC 62304 and function properly but become risky if an attacker alters the dosing logic by exploiting unmanaged vulnerability.

Functional correctness is no longer a sufficient guarantee: a system can be fully compliant and, at the same time, vulnerable to external compromises. This is where IEC 81001-5-1 comes in, extending the life cycle perspective by introducing cybersecurity as an integral part of risk management.

In modern medical software, integrating IEC 62304 and IEC 81001-5-1 does not mean performing two separate processes.

Cybersecurity does not introduce a parallel life cycle, but rather a new dimension of risk within the same activities already included in the software development life cycle.

It is precisely here that the complementarity between the two standards emerges:

- IEC 62304 defines the structure of the software life cycle. It ensures that the software behaves correctly.
- IEC 81001-5-1 extends that life cycle by introducing resiliency and cybersecurity requirements. It ensures software resiliency in the real world.

The following table shows how the IEC 62304 life cycle activities are not replaced but extended from the perspective of IEC 81001-5-1.

Software lifecycle phase	IEC 62304 Approach	IEC 81001-5-1 Extension
Requirements	Functional and safety requirements	Security and Access Control Requirements
Architecture	Software architecture and component segregation	Defense-in-depth Design and Threat Modeling
Implementation	Coding guidelines and anomaly control	Secure Coding Practices
Verification	Functional testing and Risk Control Verification	Security testing, Vulnerability Assessment and Penetration testing
Maintenance	Bug fixing and Change Management	Vulnerability Management and Patch Management
Post-market activities	Problem Resolution	Continuous vulnerability monitoring and security issue management

Table 1: *How IEC 81001-5-1 extends the IEC 62304 life cycle*

Practical impact: what does this mean for medical software developers?

The integration between IEC 62304 and IEC 81001-5-1 is not just a theoretical evolution of the regulatory framework. It concretely changes the way medical software is designed, verified and maintained over time.

In the traditional model, software **requirements** are mainly functional: they describe what the system must do, how it must behave and how to verify its correctness with respect to safety objectives. In the integrated model, this is no longer enough: each requirement must also be assessed in terms of exposure to threats, possibility of abuse and unauthorized access.

The same change is reflected in software **architecture**. Traditionally, the architecture has been built to ensure modularity, maintainability, and segregation of safety-critical functions. With the introduction of cybersecurity-by-design, however, new concepts are emerging: reduction of the attack surface, isolation of critical components and management of trust boundaries become an integral part of architectural decisions.

The verification phase also evolves profoundly. Validation is no longer limited to functional testing or verification of the risk controls required by IEC 62304. Security testing, vulnerability assessment and penetration testing are becoming increasingly important activities to demonstrate not only that software works correctly, but also that it is not easily exploitable.

It is precisely at the design level that one of the most delicate aspects of the integration between the two standards emerges. A central theme in the design of critical medical systems is the possible trade-off between safety and security. In some cases, cybersecurity requirements and clinical safety requirements may conflict.

A typical example is the implementation of a device access block after a limited number of failed authentication attempts: a measure that is effective from a security point of view (IEC 81001-5-1), but potentially problematic in emergency clinical scenarios, where the delay in access can introduce an unacceptable risk to the patient (IEC 62304).

In design practice, this type of conflict is often managed through architectural and procedural solutions that balance the two requirements, for example by providing alternative access methods in critical contexts, with different levels of authentication and control.

The integration between IEC 62304 and IEC 81001-5-1 is precisely what makes it possible to make these delicate design trade-offs explicit and analyzable, supporting a structured risk assessment that simultaneously considers safety and cybersecurity throughout the life cycle of the system.

A significant transformation is also emerging in maintenance and **post-market** activities. In the traditional model, maintenance is primarily about fixing bugs, managing changes, and controlled software evolution. In the integrated model, this view extends to continuous vulnerability management, analysis of exposure to new threats, and the release of security patches with varying priority.

In **the post-market**, it is no longer just a matter of monitoring software anomalies or functional problems, but of maintaining continuous surveillance on vulnerabilities and emerging security risks. Continuous vulnerability monitoring, security issue management, and a structured update and remediation process have become an integral part of the software life cycle.

Conclusion

The relationship between IEC 62304 and IEC 81001-5-1 is not simply the evolution of two parallel standards, but a much deeper change in the way medical software is designed, maintained, and managed over time.

For years, the dominant model has been built around safety: ensuring that software works properly and does not introduce clinical risks deriving from errors or malfunctions.

The real evolution introduced by the integration between IEC 62304 and IEC 81001-5-1 is not the addition of new technical controls, but it is the transition from a software safety model to a software trust model, in which software throughout its life cycle must be:

- correct,
- resilient,
- monitorable,
- upgradeable,
- safe

For this reason, the future of medical software will not be defined by the ability to apply different standards individually, but by their integration into a single digital risk management operational framework.

IEC 62304 defines how to build reliable medical software. IEC 81001-5-1 defines how to keep it safe in a real-world, dynamic environment. Together, they define the new integrated model of medical software.

Sources:

1. IEC 62304:2006 - Medical device software — Software life cycle processes
2. IEC 81001-5-1:2021 - Health software and health IT systems safety, effectiveness and security — Part 5-1: Security — Activities in the product life cycle
3. Security-by-Design in Medical Software (IEC 81001-5-1)

Keywords: IEC 81001-5-1, IEC 62304, IEC 62443-4-1, Cybersecurity, Software Lifecycle, Software as Medical Device, Secure Development Lifecycle, Software Security, Software Safety

Author: Maria Santoro, R&D Product Specialist - Medical Devices