




Five Sure Bets for Medical Device Developers

Strong Development Methods and
Digital Technology for Success



WINDRVR



Executive Summary

As new digital technology drives advancement in a wide range of industries, the medical device segment in particular is witnessing the importance of innovation. A 2024 study by McKinsey & Company found that 67% of healthcare leaders recognize the importance of long-term technology investments.¹

And with increased investment in digital transformation and R&D, med tech leaders are focusing on modern development methods and ways to deliver more innovation — for example, shifting from manual techniques and nondigital technology to more digital processes, such as machine learning and artificial intelligence — to power medical processes and improve healthcare. According to the World Economic Forum, \$1.3 trillion was invested in technology transformation projects in 2022, an instance of more than 10% YoY growth.²

To keep up with the market, medical device companies need to invest in new development methods and digital technology. Wind River® experts recommend a strong set of actions that medical device developers can adopt to succeed in creating new technologies to advance healthcare in the digital era:

- Effectively manage risk throughout the software development lifecycle (SDLC)
- Develop an extensible platform
- Automate everything
- Proactively manage cybersecurity threats
- Adequately secure the data

1. McKinsey & Company, "Faster, Smarter, Bolder: How Midtenure CFOs Shift into a Higher Gear," 2024, www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/

2. World Economic Forum, "How Digital Transformation Is Driving Action in Healthcare," 2022, www.weforum.org/agenda/2022/09/health-information-system-digital-transformation-healthcare

1. Effectively Manage Risk Throughout the Software Development Lifecycle



The software development lifecycle, or SDLC, is a cost-effective, time-efficient process that software developers follow to design and create high-quality software. As medical device developers plan, design, and develop their software, they need to manage the risks associated with the people, systems, and assets that interface with the device throughout its lifecycle. A negative outcome can affect everything from a patient's health, life, and privacy to the operation and integrity of a medical facility. Development teams must outline and analyze each risk, prioritize those that matter, and develop an appropriate mitigation plan by creating practical plans to address them. Furthermore, teams need to consider all risks, whether the software was developed in house or obtained from a third party.

IEC 62304

One key risk lies in medical device functional safety and the ability to obtain safety certification to protect patients, doctors, and the medical system. Advances in medical device technologies enable more effective diagnosis, monitoring, and treatment of illnesses. However, these technologies add complexities to medical device development and testing. IEC 62304 sets forth comprehensive guidelines for development and maintenance of software in medical devices. Having been adopted by both the United States and the European Union, IEC 62304 guidelines have become a standard medical device regulatory requirement and a benchmark for compliance. It establishes a set of processes, activities, and tasks within a common framework to ensure the safety and effectiveness of the medical device software development lifecycle pertaining to three different classes labeled A, B, and C.

IEC 62304 Medical Device Classifications

Class A

Impact: Minor

No possibility of damage to health

Class B

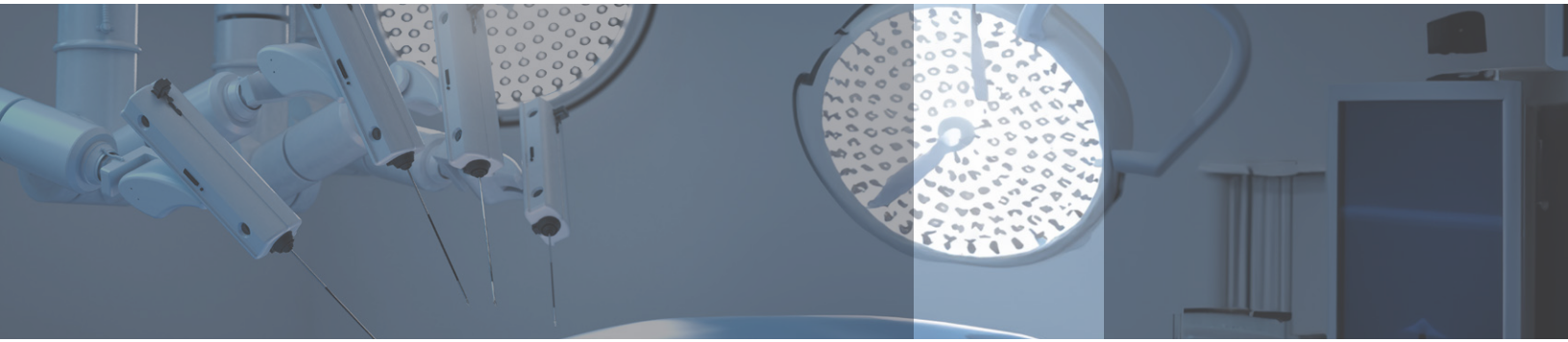
Impact: Moderate

Possibility of nonserious injury

Class C

Impact: High

Possibility of death or serious injury



Software of Unknown Provenance (SOUP)

In addition to understanding the medical device SDLC processes and compliance requirements (i.e., IEC 62304), developers need to know how to deal with SOUP — software of unknown provenance. This is software that wasn't written by the developer or device team but is generally available and usable, such as open source software, Linux, a utility software, or commercial software libraries. In most cases, this software has not been developed specifically for use within a medical device system. Adequate documentation of the development standards is not available to support and satisfy the standards set by IEC 62304, making it difficult to ensure that the software will meet compliance requirements. Utilizing SOUP as part of the medical device system calls for the development team to consider, account for, and work through the following:

- **Application use case:** How this software will be used with the medical systems
- **Robustness:** How the software enhances or changes the device operation
- **Lifecycle management:** How to manage bugs and updates supplied from a third party
- **Failure:** How to address the failure modes of the software
- **Community development process:** How updates and changes are made for open system software
- **Defect management:** The process provided by the software developers for managing defects
- **CVE management:** Upstream management of common vulnerabilities and exposures (CVEs) when they are newly discovered
- **Component management:** Managing different components of the software to meet requirements
- **Obsolescence/EOL:** Understanding and managing the full lifecycle of the software through end-of-life (EOL)

Including use of SOUP guidelines as part of the medical device software development process requires significant effort, time, and resources but helps ensure that the device complies with IEC 62304 requirements.

Developers need to know how to deal with SOUP — software of unknown provenance. This requires significant effort, time, and resources but helps ensure that the device complies with IEC 62304 requirements.



2. Develop an Extensible Platform

Next, developers should concentrate on developing an extensive and extensible platform for any medical device. It is not enough to simply develop and release the software with no additional work until end of life; instead, careful planning is required to outline what the device should accomplish throughout its lifecycle. An embedded systems journey is similar to a Linux or open source development journey: From the beginning, as part of the ideation process, a developer must consider needed work with the open source community, licensing, long-term support, security fix updates, and defect management for the device lifecycle.

After the product idea is fleshed out, the developer should create a prototype of the medical device, considering and outlining the answers to questions in at least four distinct areas:

Prototyping

- **Evaluate design choices** through extensive research, to test and validate hypotheses that answer:
 - What key design choices are needed for the medical device's entire life?
 - What are the technical and functional risks?
 - What are the approach and test risks of a hypothesis methodology?
 - What are the design decisions to be made?
 - How will we evaluate whether the correct decisions are being made?
- **Prove out new technologies** by conducting market analyses and building a business case to address:
 - What are the new technologies that can aid development of the new medical device?
 - Which do we want to use? Is there a market fit?
 - How will we determine whether the technology is right for the device?
 - How will the inclusion of the new technology impact budget or cost considerations?
 - Will using the new technology fit within the development timeline?
- **Fine-tune timeline accuracy** to communicate it to internal stakeholders and external customers, and to mitigate delivery risks, by asking:
 - What are all development and technology elements to consider?
 - Can the software development work be completed to meet the target time-to-market date?
 - How do internal and regulatory processes impact time-to-delivery?
- **Reduce overall time-to-market** by iteratively and proactively planning ahead to determine:
 - What are the key areas of risk?
 - How can they be resolved in a manner that will bring the device to market on time?
 - What resources, including cross-functional support, are needed throughout the full SDLC?



Decision: Build or Buy the OS

A major decision is whether the development team will focus its engineering resources on the medical solution's differentiators or on maintaining the operating system (OS). Questions to consider when determining whether to build or buy the OS for the medical device include:

- Are we ready to commit to supporting our OS for 10+ years?
- How will we deal with the huge number of CVEs reported each year?
- What kind of staff is required to support the OS?
- How to comply with export and compliance requirements?
- How to deal with the accelerating rate of change?
- What if the development team needs help?

Evaluating these questions will assist your decision to build your own (utilizing open source Linux) or buy a commercial Linux or real-time OS.

Navigate and Manage Technical Debt

Technical debt occurs when product development teams prioritize speed of delivery over quality for specific functionality. The results are generally issues with the code or increased difficulty in maintaining the software, and both ultimately increase overall costs. Consider these best practices when navigating technical debt:

- **Free isn't always free:** There is a total cost of ownership that comes with the use of open source software. The trade-offs of maintaining and managing open source or free software must be considered.
- **Determine the long-term impact:** An incomplete understanding of future requirements or needs can result in overlooking important issues that will require additional work or expense at a later date.
- **Build in buffers for strategic analysis and alignment:** Tight deadlines and pressure can result in decisions and actions that later cause technical debt.
- **Don't skimp on planning:** Unexpected requirements changes and scope creep cause delays, increasing rework and driving costs higher.



3. Automate Everything

An increased focus on automation can help develop medical devices and other new products. Automation has uncovered long-known but often unaddressed development issues, such as time wasted waiting for tests, builds, and pipeline debugging. When it comes to expertise and infrastructure, only 4% of employees consider their organizations expert in CI/CD,³ while 86% of developers indicate they are experiencing challenges with software complexity.⁴

Software Complexity

Software complexity combined with the use of a development infrastructure can create challenges for device developers. Using development automation can help solve this problem, and it must be built in from the beginning of the project.

The complexities that development teams need to address include:

- **Algorithmic complexity:** This requires more expertise and time for development.
- **Legacy and open source software:** The team might need to develop around older software or ensure that the open source code is up-to-date and/or secure.
- **Data analytics and AI/ML usage:** Expertise is needed to incorporate these into the medical system.
- **Development environment and size of the organization:** These can be complex, with multiple developers or development teams working on the project or on a project component.
- **Evolving security threats:** Since these are a daily concern and priority, security must be built into the development cycle at the start and constantly monitored throughout the product lifecycle.
- **Software and device testing:** Testing provides great value, but it adds to development timeline and complexity.

Automation Advantages

Automation in software development eliminates manual processes, minimizes errors, and makes development faster and more efficient overall. It also improves team success by incorporating agile best practices such as CI/CD and DevSecOps. Best-in-class organizations focus on automated builds and testing, and software automation pipelines are the cornerstone of their automation.

- A continuous integration (CI) pipeline automatically builds and tests code changes to ensure compatibility with the existing code base. A continuous deployment (CD) pipeline automates the deployment of validated and tested code changes to staging or production environments.
- DevSecOps is the collection of ideas and practices that assist the workflow automation of CI/CD. It calls for all aspects of software and device use cases to be tested for performance and efficiency. However, it can add to the complexity of the development process and timeline if not implemented effectively.

3. Globenewswire, "New Research Reveals Current Truths About the State of DevOps for Hybrid Cloud," 2021, www.globenewswire.com/news-release/2021/08/25/2286392/0/en/New-Research-Reveals-Current-Truths-About-the-State-of-DevOps-for-Hybrid-Cloud.html

4. Gartner, "What Edge Computing Means for Infrastructure and Operations Leaders," 2018, www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders

4. Proactively Manage Cybersecurity Threats



Software processes are becoming a top priority for device makers, lawmakers, and regulators. New safety and security requirements and guidelines issued in 2022 and 2023 call for medical device companies and their developer teams to ensure that cybersecurity is part of the software and device throughout the full product lifecycle.

For example, the software bill of materials (SBOM) is no longer optional. Solution providers should know the source of every file used to create their product. The SBOM has become a critical security imperative, and the government mandate Executive Order 14028 makes that clear. SBOMs are important for the company's supply chain, providing visibility as software — both open source and internally developed code — traverses the supply chain.

Another way to proactively monitor security threats comes through using the Common Vulnerability Exposure (CVE) system to identify, assess, and deliver early notification of and fixes for software threats. Monitoring through the CVE system lowers overall project costs.

Securing Your Device Is a Full Lifecycle Activity

For a medical device development team to proactively secure a device from cybersecurity threats, it must begin the process at initial development planning and continue through device decommissioning. Here are the important proactive steps and activities to consider for each stage:

- **Plan:** The initial security plan should include the data that the medical device will create and utilize for the patient, medical team, and device maker. Work on the security efforts and alignment necessary to meet compliance requirements. Determine the involvement and capacity needed from the team.
- **Develop:** Work to secure the device hardware — for example, hardening the operating system, securing system configurations, and examining the integration of additional security packages.
- **Deploy:** Prior to deployment, build in ongoing security vulnerability identification and remediation to mitigate manual effort and risks. Similarly, work on device provisioning for safe and secure setup and require a secure boot for the system.
- **Operate:** A security update plan must be in place when the medical device is installed and operating. The team should address any new requirements with signed updates.
- **Decommission:** Plan for the decommissioning of a product with steps to securely remove the device from operation. Delete sensitive data from the device and work to disable any possible attack vectors.

Know Your Security Threat Landscape

To secure your device from threats, it is important to know your security threat landscape. Consider the following:

- What needs protection?
 - Data
 - People
 - Device
 - Environment
- How are you informed?
 - CVE notifications
 - Periodic reviews
 - Ongoing scans
- How do you respond?
 - Proactive
 - Just-in-time
 - Crisis
- What is the risk tolerance?
 - Harm
 - Data loss
 - Operational disruption
 - Total failure
 - Reputation
 - Financial impact
- How do you identify your risks?
 - Asset identification
 - Threat determination
 - Vulnerability identification
- Who responds?
 - In-house team
 - Community
 - Contractors
 - Hardware providers
 - Software vendor

5. Adequately Secure the Data

Data is one of the most important outputs of today's devices, and it is critical that it be adequately secured throughout the device lifecycle. This can be done as shown in the CIA triad, representing the areas of *confidentiality*, *integrity*, and *availability*. Creating new and innovative medical devices and technology requires utilizing modern, comprehensive, and strong software development methods and digital technology.

The five development strategies highlighted in this document should assist in the device certification process and help medical device development teams create devices that are safe, secure, and reliable throughout the entire product lifecycle.

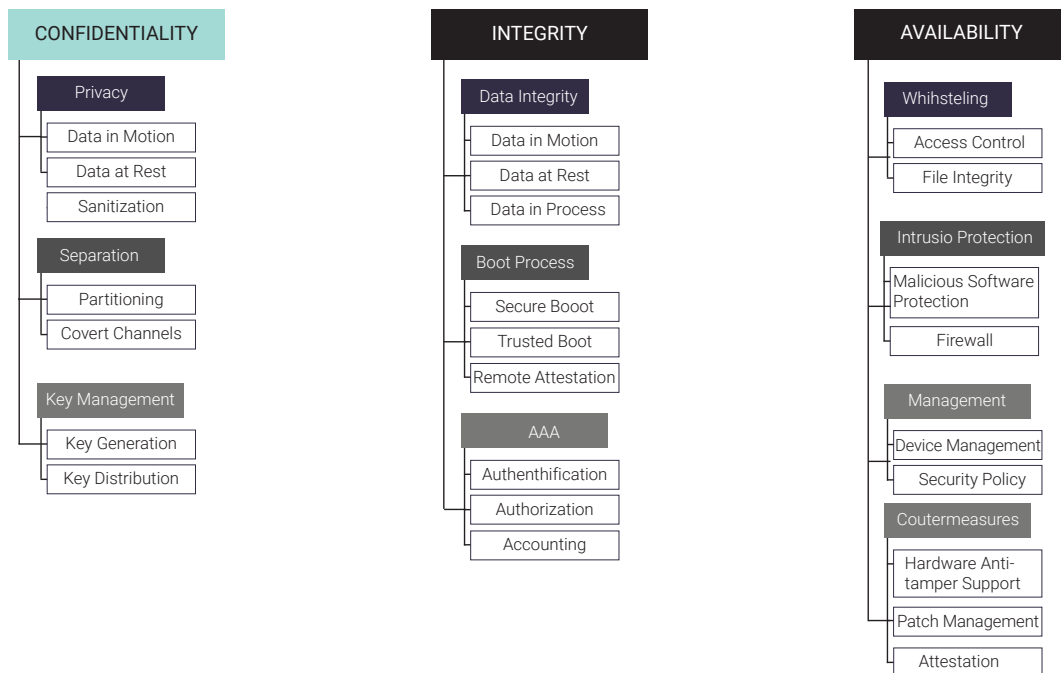


Figure 1. The CIA triad

These different security implementations can be layered together to protect the identified assets — in this case, the data from identified threats. While there are many different areas of security management, patch management and cryptographic sanitization are two that have grown in importance and focus with new data protection and data management regulations.

- **Patch management plan and system:** This must be in place to provide easy updates to a medical device to protect the data, system, medical facility, and, most importantly, the patient. This is a high priority met by recent regulatory requirements for data protection and updates to the device and software system.
- **Cryptographic sanitization:** This allows all devices to erase all site-specific data, whether that is patient data or proprietary device information. The ability to wipe the data clean prior to retirement of the device or system, so that cryptographic keys are not retrievable, will become increasingly important for the protection of patients and medical facility systems.



Conclusion

Wind River Solutions

Wind River® products and services help overcome the challenges of developing modern medical technology that is safe, secure, and reliable throughout the lifecycle.

[>> Learn More About Wind River Solutions for Healthcare Devices](#)

VXWORKS

VxWorks® supports the cost-effective development of the most advanced real-time systems, reducing time-to-market and driving innovation, new business, and revenue. Real-time and deterministic, it is backed by a DevSecOps solution and supports embedded software deployment using OCI container technology. VDC Research has ranked VxWorks as the **#1 RTOS for mission-critical real-time systems**.

[>> Explore VxWorks](#)

WIND RIVER LINUX

The market leader in commercial embedded Linux, Wind River Linux enables you to develop, deploy, and operate robust, reliable, and secure embedded solutions running on a purpose-built Linux operating system.

[>> Explore Wind River Linux](#)

[>> Explore Wind River Studio Linux Services](#)

WIND RIVER STUDIO DEVELOPER

Wind River Studio Developer is a modern DevSecOps platform that accelerates development, deployment, and operation of robust mission-critical embedded systems for the intelligent edge. It is built to solve challenges, including cost, long development cycles, limited hardware availability, isolated environments, and expensive maintenance.

[>> Explore Studio Developer](#)

WIND RIVER HELIX VIRTUALIZATION PLATFORM

Wind River Helix™ Virtualization Platform provides a single edge compute platform for running multiple applications with mixed levels of criticality. Reduce development complexity and risk and streamline the path to certification.

[>> Explore Helix Platform](#)

Services for Developers

WIND RIVER PROFESSIONAL SERVICES

Get support for medical device development at all stages, with lifecycle services, BSP development and support, Android support, Zephyr support, and more.

[>> Explore Professional Services](#)

WIND RIVER STUDIO SECURITY SERVICES

We will construct a comprehensive cybersecurity plan for your product, covering assessment, response, certifications, training, and more.

[>> Explore Studio Security Services](#)

WIND RIVER STUDIO EDUCATION SERVICES

Upgrade your team's skills, with our options ranging from on-demand learning to mentoring and custom training.

[>> Explore Studio Education Services](#)

INTEL® SIMICS

Enhance your DevSecOps process, create digital twins of complex medical systems, run virtual cybersecurity tests, and more.

[>> Explore Simics](#)

Intel and Simics are trademarks of Intel Corporation or its subsidiaries.

Wind River is a global leader of software for the intelligent edge. Its technology has been powering the safest, most secure devices since 1981 and is in billions of products. Wind River is accelerating the intelligent transformation of mission-critical edge systems that demand the highest levels of security, safety, and reliability.

© 2024 Wind River Systems, Inc. The Wind River logo is a trademark of Wind River Systems, Inc., and Wind River and VxWorks are registered trademarks of Wind River Systems, Inc. Rev. 05/2024