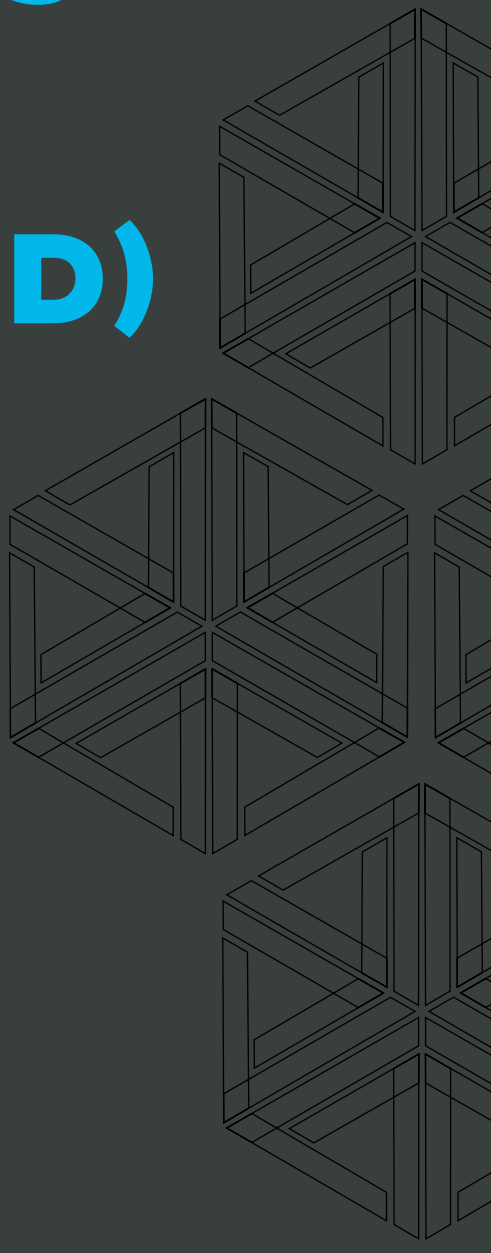
SPK and Associates

EBOOK

ULTIMATE GUIDE TO SOFTWARE AS A MEDICAL DEVICE (SAMD)

WADE SHROEDER,
MEDICAL DEVICE GURU, GREENLIGHT GURU



greenlight guru

ULTIMATE GUIDE TO SOFTWARE AS A MEDICAL DEVICE (SAMD)

TABLE OF CONTENTS

OVERVIEW

It's been more than a decade since Marc Andreesen famously wrote that "software is eating the world." Since then, software has made its way into practically every industry, adding both new conveniences and new layers of complexity to the production and distribution of millions of products.

The medical device industry is no exception. Initially, software was used to drive hardware devices. But it didn't take long before people began designing software solutions without any direct relationship to hardware devices at all. These products, while nothing like traditional medical devices, still have the potential to improve the quality of life for patients everywhere.

They are, however, still categorized as medical devices by regulatory bodies around the world. We know them as software as a medical device (SaMD), and in this guide, I want to help demystify these devices and offer you insight into how they're regulated and what you can expect as you set out to build one.

Let's take it from the top.

WHAT IS SOFTWARE AS A MEDICAL DEVICE (SAMD)?

The International Medical Device Regulators Forum (IMDRF) defines SaMD as “software intended for one or more medical purposes that perform those purposes without being part of a hardware medical device.”

FDA defines SaMD as “Software that meets the definition of a device in 181 section 201(h) of the FD&C Act and is intended to be used for one or more medical purposes without being part of a hardware device.”

While these definitions are a great starting point, there’s plenty of nuance around what exactly SaMD is and how you know if your product is SaMD. So, let’s take a closer look at what software as a medical device is, what it isn’t, and how you can figure out if your product fits the definition.

HOW DO I KNOW IF MY PRODUCT IS SAMD?

FDA is a member of the IMDRF, and it’s clear that both definitions of SaMD share a similarity in structure. According to both FDA and IMDRF definitions, there are two points that need to be fulfilled for software to achieve the status of SaMD.

To start, we need to consider whether the software can be characterized as a medical device at all. The IMDRF simply states that it must be “intended for one or more medical purposes”, while FDA specifically references the definition of a device in 181 section 201(h) of the FD&C act, which states that a device is:

An instrument, apparatus, implement, machine, contrivance, implant, in vitro reagent, or other similar or related article, including a component part, or accessory which is:

1. *recognized in the official National Formulary, or the United States Pharmacopoeia, or any supplement to them,*
2. *intended for use in the diagnosis of disease or other conditions, or in the cure, mitigation, treatment, or prevention of disease, in man or other animals, or*
3. *intended to affect the structure or any function of the body of man or other animals, and which does not achieve its primary intended purposes through chemical action within or on the body of man or other animals and which does not achieve its primary intended purposes through chemical action within or on the body of man or other animals and which is not dependent upon being metabolized for the achievement of its primary intended purposes. The term "device" does not include software functions excluded pursuant to section 520(o).*

To appropriately use this definition, you need to define your product's intended use and its indications for use. As a refresher:

- **Intended use** is the purpose of your device. It's what your device will be used for.
- **Indications for use** are the diseases or conditions that your device will diagnose, treat, prevent, cure, or mitigate. Indications for use describe who your device will be used on and *why*.

Once you've defined these uses, points two and three in the FDA's definition of a medical device should make it fairly clear whether your product will be regulated as a medical device.

If you plan on marketing your software product in the US, I highly encourage you to read carefully through this FDA guidance on Policy for Device Software Functions and Mobile Medical Applications. This guidance offers clear stances on what software functions FDA considers to be a medical device, as well as those it does not consider to be medical devices, or those which it does not regulate as such.

However, if you're still not sure if your product is a medical device, your best bet is to contact FDA directly.

IS MY SOFTWARE CONSIDERED SAMD OR SIMD?

Let's say you've determined that your product meets the definition of a medical device. You still need to consider the second half of the SaMD definitions from the IMDRF and FDA.

- The IMDRF definition tells us that the software must perform its purposes "without being part of a hardware medical device."
- FDA uses almost the exact same language, stating that SaMD "is intended to be used for one or more medical purposes *without being part of a hardware device.*"

This narrows the scope of SaMD again. Software that is used to power hardware or drive a hardware device does not meet the criteria for SaMD.

Instead, that type of software is what's known as SiMD, or "software *in* a medical device."

Simply put, any software that helps to run a hardware medical device—say, by powering its mechanics or producing a graphical interface—is software in a medical device. Some examples include:

- Software that controls the inflation or deflation of a blood pressure cuff
- Software that controls the delivery of insulin on an insulin pump
- Software used in a closed loop control of a pacemaker

These types of software are also known as "embedded software" of "firmware", or "micro-code" so if you hear those terms, know that they indicate SiMD, not SaMD.

For a product to be classified as SaMD, the software must stand alone from any hardware as it performs the functions that categorize it as a medical device. IMDRF guidance on Possible Framework for Risk Categorization and Corresponding Considerations adds several clarifying points to the SaMD definition:

- SaMD is a medical device and includes in-vitro diagnostic (IVD) medical device.
- SaMD is capable of running on general purpose (non-medical purpose) computing platforms.
- "without being part of" means software not necessary for a hardware medical device to achieve its intended medical purpose.

- Software does not meet the definition of SaMD if its intended purpose is to drive a hardware medical device.
- SaMD may be used in combination (e.g., as a module) with other products including medical devices.
- SaMD may be interfaced with other medical devices, including hardware medicaldevices and other SaMD software, as well as general purpose software.
- Mobile apps that meet the definition above are considered SaMD.

NOTE: While it's important to differentiate between SaMD and SiMD, the two types of software share many of the same standards for development, such as IEC 62304, the international standard for software lifecycle processes. If your software is actually SiMD, you'll still find many of the guidance documents and standards in this guide useful.

WHAT ARE SOME EXAMPLES OF SAMD?

Theory is one thing, but it's often easier to grasp real examples. I'll go over a few examples of SaMD here, but you can find more in the same IMDRF guidance on risk classification mentioned above:

- Software that allows a mobile device to view images from an MRI, ultrasound, or X-ray that are used for diagnostic purposes.

- Software that processes images to help detect breast cancer.
- Software that diagnoses a condition using the tri-axial accelerometer on a smartphone.
- Software that collects patient data in real-time that is monitored by a medical professional and used to develop treatment plans.

Additionally, here are a few real-world examples of Greenlight Guru customers whose products are categorized as SaMD.

- Avatar Medical creates 3D images of patients based on their medical images to help surgeons better visualize the images.
- BrainKey uses MRI scans to create a 3D image of your brain that you can track over time as you get more MRI scans.
- Brain+ is an app that treats dementia by cognitive stimulation.
- Cloud of Care's software uses AI to increase the certainty and efficiency of long-term EEG readings in epilepsy.

HOW IS SAMD REGULATED AROUND THE WORLD?

While there are many markets for medical devices around the world, the US and the EU are by far the largest, so these are the markets we'll focus on in this section.

The first point that I want to make is that a SaMD product is still a medical device, and is regulated as such.

For starters, you will need a quality management system (QMS). In the US, you must follow the Quality System Regulations (QSR) from the FDA. Likewise, in the EU, your SaMD will be governed by the EU MDR (or EU IVDR if it is an in vitro diagnostic device). And finally, your device will still be classified according to the applicable regulations in either market.

Basically, it pays to remember that while your SaMD may be significantly different from a traditional, hardware medical device, you still need to follow the same regulations as any other medical device.

With that in mind, let's get into some of the nuances of the SaMD regulatory landscapes in the US and EU.

WHAT YOU NEED TO KNOW ABOUT SAMD REGULATION IN THE US

FDA recognizes its medical device regulations were written with traditional medical devices in mind, which is why they've since released guidance documents specific to software for areas like premarket submissions.

Its first guidance on premarket submissions for SaMD was published back in 2005. If you're thinking it might be a little dated, you're not wrong. That's why FDA drafted a new guidance for premarket submissions for SaMD in 2021.

This is where things get tricky. Both the current guidance and the draft guidance list the documentation you need to submit based on the intended use of your SaMD.

However, the current guidance (from 2005) divides SaMD into three categories, known as “levels of concern”, which are based on the severity of injury that could arise from device failure or design flaws:

- **Minor**—failures or latent design flaws are unlikely to cause any injury to the patient or operator
- **Moderate**—failures or latent design flaws could directly result in minor injury of the patient or operator, including through delayed or incorrect information or through the actions of a provider
- **Major**—failures or latent design flaws could directly result in death or serious injury to the patient or provider, including through delayed or incorrect information or through the actions of a provider

The current guidance document then specifies the documentation you’ll need to submit is based on the level of concern your device falls under. Please note that “level of concern” is not the same as your device’s risk class.

Now let’s look at the draft guidance. In this case, the three levels of concern have been replaced with two levels of documentation:

- **Basic Documentation**
- **Enhanced Documentation**

As you can imagine, this has caused some confusion. We don’t know when the draft guidance will be finalized or even what the new guidance will look like

until it has been finalized. So in the meantime, I recommend that you indicate both your level of concern and whether your device requires basic or enhanced documentation.

That might sound onerous, but the truth is the draft guidance and the published guidance require similar documentation—they just categorize SaMD differently. So, the documentation you compile for one guidance will likely be the documentation you need for the other.

If you're unsure of which category your SaMD falls under in either guidance, I'd err on the side of caution and submit the documentation required for the higher level.

WHAT YOU NEED TO KNOW ABOUT MEDICAL DEVICE SOFTWARE REGULATION IN THE EU

SaMD regulation in the EU is similar to regulation in the US, in that it does not differ from the way traditional medical devices are regulated. You will still need to comply with all the relevant requirements in the EU MDR and EU IVDR.

It's important to note, however, that EU regulations do not use the term "software as a medical device." Rather, they use the term "medical device software" or MDSW for short.

Fortunately, the European Commission (EC) has put out several guidance documents relevant to SaMD manufacturers.

- MDCG 2021-24—Guidance on classification of medical devices

- Infographic—Is your software a medical device?
- MDCG 2020-1—Guidance on clinical evaluation and performance evaluation of medical device software
- MDCG 2019-16—Guidance on cybersecurity for medical devices
- MDCG 2019-11—Qualification and classification of software

I'll get into some of these later in the guide, but for now, I'd encourage you to read or at least bookmark these resources for easy reference. You'll find them immensely helpful, especially if you're wondering whether your software meets the definition of medical device software in the EU.

If you know your device is MDSW and you want to make sure you're complying with EU MDR requirements, using this free guidance document and gap assessment tool from Greenlight Guru is a great way to assess your compliance and determine the appropriate regulatory route for your medical device software.

HOW IS SAMD CLASSIFIED ACROSS GLOBAL REGULATORY MARKETS?

So far, we've learned about the regulations, guidance, and international standards that apply to SaMD.

Unfortunately, not only do the US and the EU have different risk categories for medical devices, but the IMDRF and IEC 62304 also contain methods for categorizing SaMD. This can get confusing quickly, so let's use this section to break down each category and class and how they relate to one another.

SAMD RISK CLASS AND "LEVELS OF CONCERN" IN THE US

First, we have the US system for classifying SaMD. FDA classifies SaMD using the same risk classes as it does for traditional medical devices: Class I, Class II, and Class III.

Just to reiterate, although you will have to choose a "level of concern" for your pre-market submission to FDA, this does not determine your risk class. Level of concern merely tells you the documentation your pre-market submission will require. While it may be strongly correlated with risk class, your level of concern is not used to determine your device's risk classification.

MEDICAL DEVICE SOFTWARE RISK CLASS AND "RULE 11" IN THE EU

As with the US, there is no MDSW-specific risk classification in the EU. Medical device software uses the same risk classification as traditional medical devices: class I, class IIa, class IIb, and class III.

But the EU MDR has an outline for how you should go about determining your medical device software risk class, identified as Rule 11.

Rule 11, which can be found in Annex VIII of EU MDR, states:

> *Software intended to provide information which is used to take decisions with diagnosis or therapeutic purposes is classified as class IIa, except if such decisions have an impact that may cause:*
>
> - *death or an irreversible deterioration of a person's state of health, in which case it is in class III; or*
> - *a serious deterioration of a person's state of health or a surgical intervention, in which case it is classified as class IIb.*
>
> *Software intended to monitor physiological processes is classified as class IIa, except if it is intended for monitoring of vital physiological parameters, where the nature of variations of those parameters is such that it could result in immediate danger to the patient, in which case it is classified as class IIb.*
>
> *All other software is classified as class I.*

The EC has elaborated on Rule 11 and its process for classification in MDCG 2021-24 and MDCG 2019-11. However, as you may have noticed reading Rule 11, most medical device software will be classified as at least class IIa under the rule.

SAMD CATEGORIZATION ACCORDING TO IMDRF

The IMDRF has also put out guidance on categorizing SaMD. What you'll notice immediately when you read this document is that the IMDRF categorization has

four possible categories, rather than three, and requires a table to help you identify what category your device falls under.

This table is two-dimensional, requiring the identification of both situation or condition and the significance of the information provided by the SaMD.

State of Healthcare situation or condition	Significance of information provided by SaMD to healthcare decision		
	Treat or diagnose	Drive clinical management	Inform clinical management
Critical	IV	III	II
Serious	III	II	I
Non-serious	II	I	I

"Software as a Medical Device": Possible Framework for Risk Categorization and Corresponding Considerations, IMDRF

Currently, the IMDRF categorization is not widely used, probably because it seems to add a second, confusing layer to SaMD classification. However, the IMDRF categorization can be useful in helping to determine your risk class in the EU, and this is what most companies use it for.

One of the aforementioned EU guidance documents, MDCG 2019-11, includes a table that combines both the IMDRF categorization and EU risk classes.

		Significance of Information provided by the MDSW to a healthcare situation related to diagnosis/therapy		
		High Treat or diagnose *IMDRF 5.1.1*	**Medium** Drives clinical management *IMDRF 5.1.2*	**Low** Informs clinical management *(everything else)*
State of Healthcare situation or patient condition	**Critical** situation or patient condition *IMDRF 5.2.1*	**Class III** *Category IV.i*	**Class IIb** *Category III.i*	**Class IIa** *Category II.i*
	Serious situation or patient condition *IMDRF 5.2.2*	**Class IIb** *Category III.ii*	**Class IIa** *Category II.ii*	**Class IIa** *Category I.ii*
	Non-serious situation or patient condition *(everything else)*	**Class IIa** *Category II.iii*	**Class IIa** *Category I.iii*	**Class IIa** *Category I.i*

Table 1: classification Guidance on Rule II, MDCG 2019-11

So, if you intend to sell your device in the EU, the IMDRF categorization can be a useful tool for helping you determine the risk class of your SaMD.

SOFTWARE SAFETY CLASSIFICATION ACCORDING TO IEC 62304

Finally, we have the software safety classification of IEC 62304, the international standard on software lifecycle processes. I'll get into IEC 62304 in more depth later in this guide, but for now, I want to focus on its classification system.

The IEC 62304 classification system has three levels based on the severity of injury that a software failure could cause:

- Class A—no injury
- Class B—non-serious injury
- Class C—serious injury or death

NOTE: Software safety classification is not a determination of how safe your software is. Instead, it is the basis for determining how rigorous your software development process must be.

The actual safety of your device depends on your design and development processes, not on a safety classification.

Something you should keep in mind is that your safety classification under IEC 62304 does not directly correspond with risk classification under US or EU regulations. Safety classification does, however, have a strong correlation with risk class.

For example, if your safety classification is class C, then there's a good chance your device will be class III (for either US or the EU). But you could conceivably have a device that is safety class C, yet falls into a lower risk class.

The point being, the IEC 62304 classification system is about safety, but it only indicates the level of rigor you should use during software development. It is not a perfect proxy for risk classification and it does not tell you anything about the actual safety of a finished SaMD.

If you're still with me, then give yourself a pat on the back. This is tricky stuff, but depending on where you're planning to market your device, you may need to understand how to use every one of these classification methods.

Perhaps the most important takeaway here is that you should never assume that one classification method, such as level of concern or software safety class, will directly correspond with another.

SOFTWARE DEVELOPMENT FOR SAMD

Start talking about software development in the medical device world, and you're likely to hear a lot of pointed opinions.

The big reason for this is simple. A lot of regulations, like FDA's QSR, were

written with traditional, hardware medical devices in mind. As such, they assume a very linear style of product development, where one task is accomplished before you proceed to the next. This is generally known as the waterfall methodology, and because regulations and standards like IEC 62304 are written this way, it causes a lot of angst among software developers.

Most developers today use an agile methodology, a more flexible approach based on a continuous loop of iteration. On top of that, many emerging SaMD companies don't come from the traditional medical device world. As a result, many of these teams think it's impossible to apply an agile methodology in medical device development based on current regulations.

So let me set the record straight right here and now. Yes, the regulations are written in a linear style that easily tracks with a waterfall approach. However, it is very much still possible to use agile product development and still maintain compliance with all the relevant regulations and standards.

For instance, when you first look at IEC 62304, you'll notice that it has a very linear structure. But it is still possible to fulfill its requirements while using agile product development. In fact, there is another standard AAMI TIR 45 that offers guidance on the use of agile practices in the development of medical device software.

With that out of the way, let's take a closer look at IEC 62304 and how it's used in SaMD development.

IEC 62304—LIFECYCLE REQUIREMENTS

IEC 62304 is a process standard, as opposed to a product standard. It will not tell you specific requirements for your product. Instead, it offers a method for structuring your processes that will result in safe and effective software, if carried out properly.

After the general requirements, IEC 62304 explains the five processes you'll need to follow during the lifecycle of your software:

- Software development process
- Software maintenance process
- Software risk management process
- Software configuration management process
- Software problem resolution process

Keep reading for a brief description of each of these processes, but I also encourage you to check out this course on Greenlight Guru Academy that offers a thorough, but highly accessible introduction to IEC 62304.

Software development process

The software development process, according to IEC 62304, begins with software development planning and ends with the software release. Between those points, you'll need to carry out a number of essential steps, including:

- Software requirements analysis
- Software architectural design

- Software unit implementation and verification
- Software integration and integration testing
- Software system testing

However, once the software has been released, you're by no means finished with your responsibilities. Remember, IEC 62304 is a software lifecycle standard, which means you'll need to maintain the software and resolve problems as they arise.

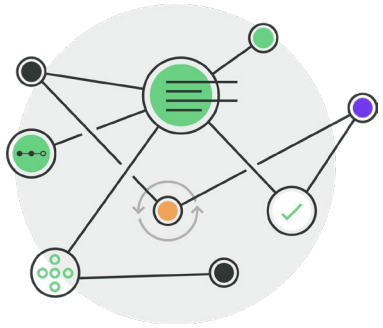
Software maintenance process

IEC 62304 lists the requirements for a software maintenance process. These requirements will look very similar to complaint handling requirements from ISO 13485 and FDA 21 CFR Part 820.

The software maintenance process outlined in IEC 62304 consists of three parts:

- Establishing your software maintenance plan
- Analysis of problems and modifications
- Implementing modifications

TIP: Study your existing complaint handling process before you create a new process for software maintenance. There may already be a lot of overlap between the two, and you may not need two separate processes.



With Greenlight Guru's dedicated Complaint Management software, feedback and complaints are captured in the same single system that handles risk management, giving you confirmation that you've captured the appropriate risks and acceptability. Because you can link anything to anything else in the system, you can see everything that might be tied to or impacted by the feedback.

Software risk management process

If you're coming to IEC 62304 from a medical device manufacturing background, then the software risk management process it outlines should look pretty familiar to you. That's because the risk management requirements in IEC 62304 correlate with those in ISO 14971, the international standard on the application of risk management to medical devices.

And remember, SaMD manufacturers are expected to follow ISO 14971, just like any other medical device manufacturer. If you're unclear about anything related to risk management, then check out our Definitive Guide to ISO 14971.

You'll quickly notice the connection between ISO 14971 requirements and those of IEC 62304, such as:

- An analysis of software contributing to hazardous situations
- Risk control measures

- Verification of risk control measures
- Risk management of software changes

Keep in mind, just because risk management gets its own section, that doesn't mean it isn't relevant to other processes. In fact, the software development process itself can be used as a risk control measure when it's carried out according to IEC 62304.

Software configuration management process

Software configuration is like accounting for your software. You're keeping track of everything you would need to recreate the software, which is foundational for traceability and release management.

The requirements laid out in IEC 62304 include:

- Configuration identification
- Change control
- Configuration status accounting

Many companies use a configuration matrix to help them with configuration management. This matrix combines both the software items you want to control and when they were released in one handy table, like the one illustrated below.

	Release 1	Release 2
Software development plan	Rev. 001	Rev. 002
Source code	Rev. 001	Rev. 002
Build environment specification	Rev. 003	Rev. 003
Software requirement	Rev. 002	Rev. 002
Software verification specification	Rev. 004	Rev. 004
Document XXX		
...		

Software problem resolution process

The term “software problem resolution process” is a bit of a misnomer, because it implies that you need one problem resolution process. In reality, you will need several different problem resolution processes because you’ll encounter different types of problems throughout the software lifecycle.

IEC 62304 does lay out a general outline for a problem resolution process:

- Prepare problem reports
- Investigate the problems
- Advise relevant parties
- Use change control process
- Maintain records
- Analyze problems for trends
- Verify software problem resolution
- Test documentation contents

As a rule of thumb, you'll likely encounter fewer problems over time as your product matures. However, the severity of the problems you encounter later on are likely to be more severe—and will require a more rigorous process for resolution.

SOFTWARE VALIDATION FOR SAMD DEVELOPMENT

According to the FDA's Quality System Regulations,

> *When computers or automated data processing systems are used as part of production or the quality system, the [device] manufacturer shall validate computer software for its intended use according to an established protocol.*

For those coming from a software development background, without much insight into the medical device industry, the requirement to validate software that's used to build other software may be an unexpected speed bump. However, this type of validation is essential to producing safe medical devices, whether they're hardware or software.

How much validation is required? Well, in FDA's guidance on the General Principles of Software Validation, it states, "The level of validation effort should be commensurate with the risk posed by the automated operation."

For instance, if the software application that you're using for testing doesn't work correctly and gives you a false pass, or doesn't test everything it should have,

then you're looking at a serious issue that could affect patient health and safety.

On the other hand, you don't need to validate something like Microsoft Excel for general use. It's a widely used product that poses little risk to your product.

Just remember that all of your validation must be done in accordance with a documented protocol. And the results of that validation must be documented, as well.

Ultimately, the decision about whether or not to validate a software tool comes down to you. Keep in mind, however, that whatever your decision, you will be expected to justify it.

At Greenlight Guru, we know how time consuming it can be to validate software tools for medical device development. That's why we do it for you. Every major update of our software comes with a complete validation package so you can be sure you're compliant while you're developing your devices.

CYBERSECURITY AND SAMD

Safety is always paramount when it comes to medical devices. And creating safe SaMD means considering an extra element: cybersecurity.

Whether you're coming from the software development world or the medical device industry, the need for cybersecurity measures should be clear by now.

There have been a number of high-profile attacks in the healthcare industry in the past decade, and new vulnerabilities are being discovered all the time. In fact, healthcare is regularly cited as one of the most at-risk industries for cyberattacks.

All of this means that device makers can no longer afford to put cybersecurity on the backburner or try to add it into a finished device. The threats are real, and it's critical that you take them seriously to protect the safety of your device users.

SOFTWARE BILL OF MATERIALS (SBOM) FOR SOFTWARE AS A MEDICAL DEVICE

On May 12, 2021, the Biden administration issued the Executive Order on Improving the Nation's Cybersecurity. One important aspect of this executive order was its focus on a software bill of materials (SBOM).

An SBOM is a nested inventory of all third party software that exists within SaMD or SiMD. A software bill of materials is crucial to the safety and security of your product, because it provides a list of ingredients for your device.

If a vulnerability is found in some widely used software component, you can quickly check whether that component is in your list of ingredients. If it is, you'll know which products are affected and can quickly alert providers and work to fix the issue.

The executive order directed the National Telecommunications and Information Agency (NTIA) to create a list of the minimum elements that are required for an SBOM, which the NTIA has since published. The minimum elements include:

- **Data Fields—**Document baseline information about each component that should be tracked: Supplier, Component Name, Version of the Component, Other Unique Identifiers, Dependency Relationship, Author of SBOM Data, and Timestamp.
- **Automation Support—**Support automation, including via automatic generation and machine-readability to allow for scaling across the software ecosystem. Data formats used to generate and consume SBOMs include SPDX, CycloneDX, and SWID tags.
- **Practices and Processes—**Define the operations of SBOM requests, generation and use including: Frequency, Depth, Known Unknowns, Distribution and Delivery, Access Control, and Accommodation of Mistakes.

Now, you may be looking at these minimum elements and thinking that this looks like a lot of work to put together. And you're not wrong—it's true that compiling an SBOM on your own and monitoring every component for new vulnerabilities is a heavy lift. This is why I'd recommend using an automated SBOM tool that will compile all of this for you and proactively monitor the elements in your SBOM for new vulnerabilities.

There are a number of tools out there for you to use, including the SBOM Analysis and Vulnerability Management Tool from our partners at MedCrypt. I'd encourage you to explore your options before building your SBOM manually.

The SBOM requirements are new, and I know there's always some initial resistance to a brand new requirement, but it really is critical (and expected) that you have an SBOM for your software as a medical device.

CYBERSECURITY REGULATIONS, GUIDANCE, AND RESOURCES

Cybersecurity is a relatively new and ever-evolving field for regulators and medical device professionals alike. However, regulatory bodies are catching up, publishing guidance documents and resources that every SaMD professional should read and understand.

I've compiled a short list of them here and it's a good place to start if you're wondering what regulatory bodies will expect from you and your product regarding cybersecurity:

- **Playbook for Threat Modeling Medical Devices**. Threat modeling is one of the best methods for strengthening the security and safety of your SaMD. That's why FDA has developed this threat modeling playbook. In it you'll find resources for developing and adapting your company's threat modeling practices.
- **Cybersecurity for Networked Medical Devices Containing Off-the-Shelf (OTS) Software**. This is the FDA guidance on the use of off-the-shelf (mass marketed) software in medical devices. A vulnerability in this type of software may pose a risk to the operation of medical devices and generally requires ongoing maintenance throughout the product lifecycle. This guidance clarifies how regulations like the QSR apply to cybersecurity maintenance activities.

- **Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions** (DRAFT). This FDA draft guidance has not yet been finalized. However, draft guidances represent the agency's latest thinking on a subject. And while some elements of the guidance may change before it's finalized, this draft is essential reading for SaMD manufacturers.
- **Postmarket Management of Cybersecurity in Medical Devices**. This FDA guidance document provides the agency's recommendations for managing postmarket cybersecurity vulnerabilities in medical devices. It offers specific recommendations and encourages a total product lifecycle approach to cybersecurity.
- **MDCG 2019-16—Guidance on Cybersecurity for Medical Devices**. This guidance document is meant to help manufacturers fulfill the requirements of Annex I of both EU MDR and EU IVDR, with regard to cybersecurity. It also includes the expectations from other stakeholders, such as integrators, economic operators, and users.
- **IMDRF Principles and Practices for Medical Device Cybersecurity**. This guidance was issued to offer best practices and general principles for medical device cybersecurity. The IMDRF's stated goal for the guidance is to "facilitate international regulatory convergence on medical device cybersecurity," and it offers recommendations for both device manufacturers and external stakeholders, such as providers.

For an even deeper dive, check out the cybersecurity page from FDA, which contains a lengthy list of news and updates, white papers and reports, guidance documents, known vulnerabilities, and other resources on cybersecurity.

It's also worth noting that a bipartisan bill known as the PATCH Act is currently making its way through Congress in the US. If it becomes law, it will impose a number of cybersecurity requirements for manufacturers applying for premarket approval. It will also codify the need for an SBOM and require manufacturers to address postmarket cybersecurity vulnerabilities.

If there's one thing I can impress upon you about cybersecurity, it's that you need to start thinking about it early on. The expectation is that you're addressing cybersecurity throughout the design process, rather than treating it as an "extra" that's tacked on once your product is finished.

The sooner you start thinking about cybersecurity for your SaMD, the easier it will be to meet regulatory expectations (and avoid breaking any laws, should the PATCH Act pass). Most importantly, however, baking cybersecurity into the design of your device will result in a safer, more secure product for the patients and providers using it.

POSTMARKET REQUIREMENTS FOR SAMD

It bears repeating that while there are certainly nuances and added issues to deal with regarding SaMD (like cybersecurity), software as a medical device is still subject to the same regulations as hardware medical devices.

This means all of the postmarket requirements from regulations like FDA's QSR, EU MDR or IVDR still very much apply to your SaMD. Additionally, if you've been following IEC 62304 for software development, your software maintenance process and software problem resolution process will help you fulfill some of these postmarket requirements.

For a more comprehensive understanding of these regulations, check out some of the postmarket-related guides, podcasts, and webinars from our library of free medical device resources.

With that said, let's talk about some of the features inherent in SaMD that require a different approach in the postmarket stage of a medical device's lifecycle.

WHEN DOES A SAMD REQUIRE A NEW SUBMISSION?

Making a change to a medical device that is on the market will always require scrutiny. At the very least, you'll need to document the change you've made. But if a change is significant enough, it may require you to resubmit documentation you needed to get the device on the market in the first place.

When your product is software, deciding whether you need a new submission is even more complicated. So, let's take a look at how you'll make that decision.

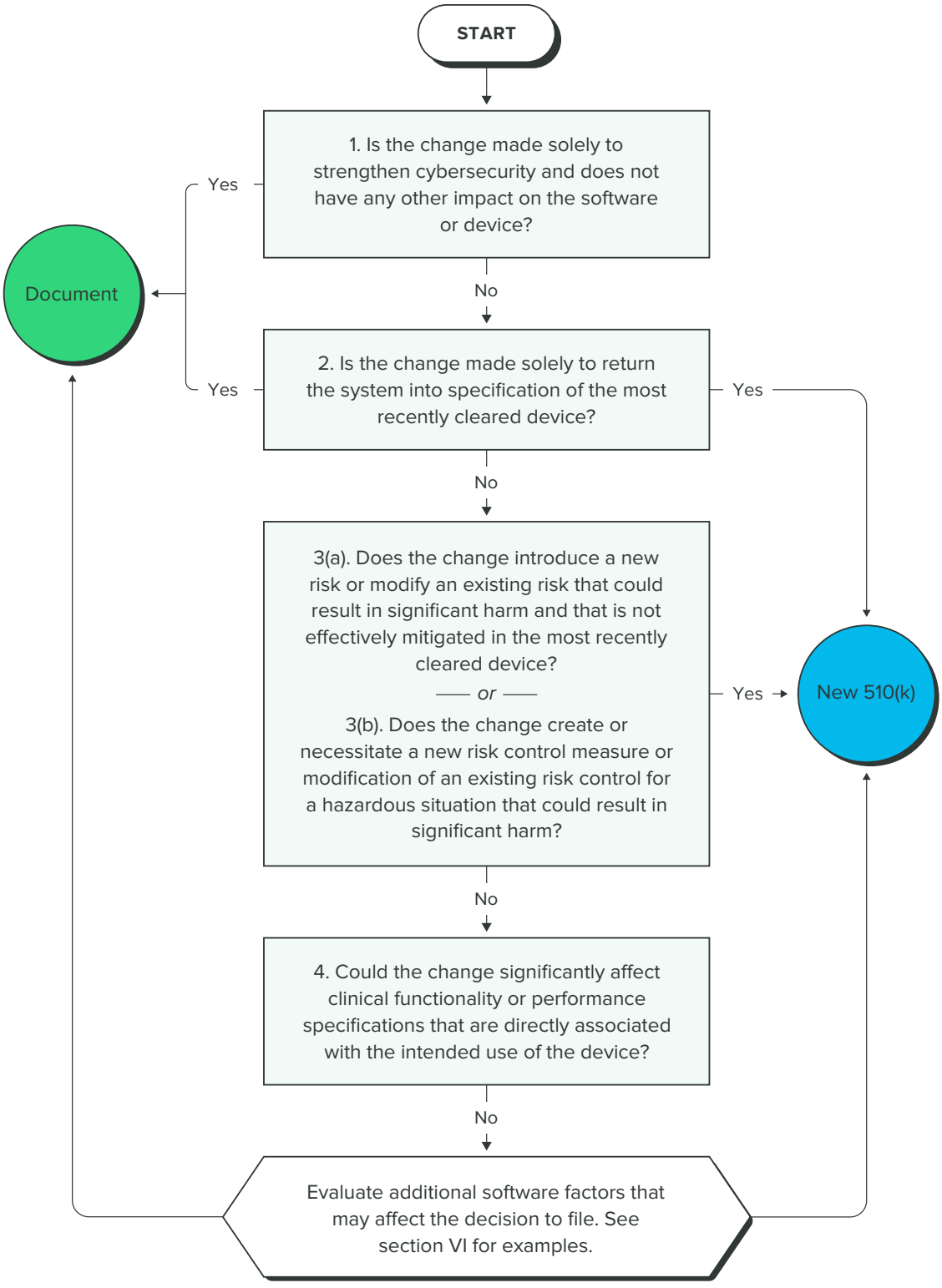
Making changes to SaMD in the US

Software as a medical device is similar to traditional medical devices in that if you want to makes changes to your device, you have two options:

- Notify FDA of the change via a new 510(k), a special 510(k), or PMA supplement
- Document the change internally via a letter-to-file

The choice you make depends on the significance of the change you're making and whether it affects the safety, efficacy, or performance of the device. For a hardware medical device, it's usually easier to know whether your change is significant enough to warrant notifying FDA. But with software, the decision can be a little trickier.

Fortunately, FDA has released a guidance on Deciding When to Submit a 510(k) for a Software Change to an Existing Device to help SaMD manufacturers. As with all the guidance documents listed in this guide, I encourage you to read the entire thing. However, this flowchart will give you an idea of the questions you should ask during the decision-making process *(see the flowchart on the next page)*.



Deciding When to Submit a 510(k) for a Software Change to an Existing Device, FDA

Making changes to SaMD in the EU

The requirements for a notification of a change in your device in the EU are similar. Annex X of EU MDR states:

> *The applicant shall inform the notified body which issued the EU type-examination certificate of any planned change to the approved type or of its intended purpose and conditions of use.*
>
> *Changes to the approved device including limitations of its intended purpose and conditions of use shall require approval from the notified body which issued the EU type-examination certificate where such changes may affect conformity with the general safety and performance requirements or with the conditions prescribed for use of the product. The notified body shall examine the planned changes, notify the manufacturer of its decision and provide him with a supplement to the EU type-examination report. The approval of any change to the approved type shall take the form of a supplement to the EU type-examination certificate.*
>
> *Changes to the intended purpose and conditions of use of the approved device, with the exception of limitations of the intended purpose and conditions of use, shall necessitate a new application for a conformity assessment.*

Although the MDR requirements are not exactly the same as the FDA guidelines, you should be able to notice a similar theme running through both: modifications that change the intended use of the device require a new submission to FDA or application to your notified body.

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING IN SAMD

There's one more wrinkle in the decision-making process for a new submission: artificial intelligence (AI).

AI is one of the fastest growing technologies being applied to medical devices. Our benchmark industry survey, the 2022 State of Medical Device Quality, Product Development, and Commercialization Report, found that a quarter of respondents are including AI in their products.

This includes machine learning (ML), a subset of AI that has exploded in use across many industries in recent years.

Machine learning refers to an algorithm that has the ability to change or improve its outputs as it "learns" from an increasing amount of inputs. The more data an ML algorithm receives, the more accurate its results can become.

This technology offers incredible potential for diagnostics and many other medical devices, but it does pose a dilemma. When we're talking about SaMD, the change in the algorithm that allows it to improve is technically a change in the device itself.

So, how do you know if that change has reached the level of requiring a new submission for your SaMD?

This is still relatively new territory for medical device manufacturers and regulatory bodies. However, FDA has released a Proposed Regulatory Framework for Modifications to Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device (SaMD) to address this issue. While I highly recommend

reading the full document, there are a couple things I want to point out here.

First, FDA still expects manufacturers to refer to and use the guidance document, *Deciding When to Submit a 510(k) for a Software Change to an Existing Device*, which we discussed earlier in this section.

Second, FDA is also proposing a framework for AI/ML modifications based on the principle of a "predetermined change control plan." Basically, this would establish the parameters of any anticipated modifications when you send in your first premarket submission.

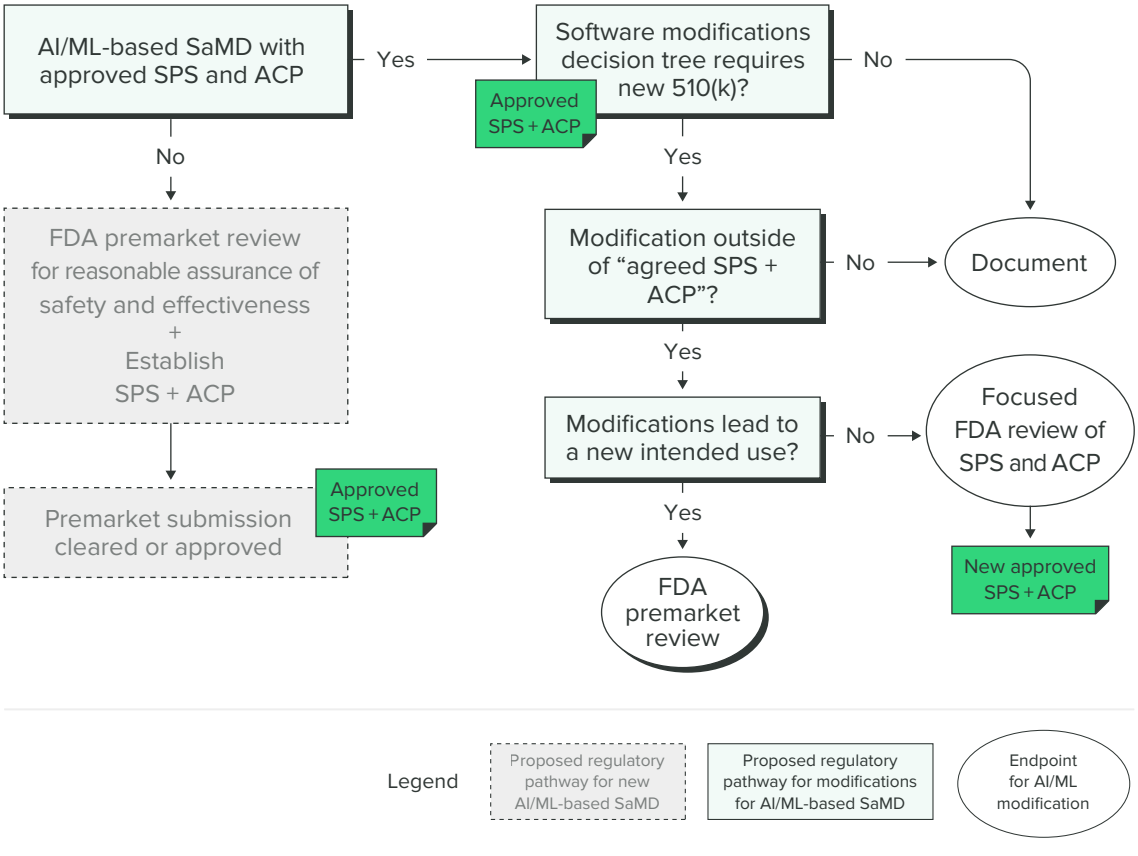
This "predetermined change control plan" takes the form of two documents outlining two different types of modifications. FDA refers to these as:

- **SaMD Pre-Specifications (SPS):** A SaMD manufacturer's anticipated modifications to "performance" or "inputs" or changes related to the "intended use" of AI/ML-based SaMD. These are the types of changes the manufacturer plans to achieve when the SaMD is in use. The SPS draws a "region of potential changes" around the initial specifications and labeling of the original device. This is "what" the manufacturer intends the algorithm to become as it learns.
- **Algorithm Change Protocol (ACP):** Specific methods that a manufacturer has in place to achieve and appropriately control the risks of the anticipated types of modifications delineated in the SPS. The ACP is a step-by-step delineation of the data and procedures to be followed so that the modification achieves its goals and the device remains safe and effective after the modification. This is "how" the algorithm will learn and change while remaining safe and effective.

Under this proposed framework, FDA suggests that modifications that fall within the agreed upon boundaries of the SPS and ACP would only need to be documented by the manufacturer. Basically, these are the modifications that you informed FDA could happen as the algorithm changes.

Modifications *outside* the boundaries of the SPS and ACP would require a new 510(k) submission if the modifications affect the safety or effectiveness of the device.

I know that's a lot to take in, so this flowchart should help you wrap your head around it.



Proposed Regulatory Framework for Modifications to Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device (SaMD), FDA

In addition to FDA's proposed regulatory framework, FDA, Health Canada, and the United Kingdom's Medicines and Healthcare Products Regulatory Agency have come together to issue guiding principles for Good Machine Learning Practice for Medical Device Development. This resource is short and to the point, and it's essential reading for anyone building a product that includes ML.

This is still a developing field, but these principles should help guide manufacturers interested in using AI/ML technologies with a SaMD.

FINAL THOUGHTS ON SOFTWARE AS A MEDICAL DEVICE

If you've made it to this point of the guide (even if you've just skimmed the headers), I think you'll agree that the world of SaMD is both complex and evolving.

There are still gray areas within regulations and guidances for SaMD that will need to be addressed in the coming years. There is still much work to be done around issues like cybersecurity. And there's a steep learning curve for software developers entering the medical device world, and vice-versa.

But there's also immense opportunity and excitement in this space. The last thing I want is for you to finish this guide and be put off by the work that goes into

building safe and effective software as a medical device. With the right tools, and the best expert advice on hand, there's no reason why your company can't create high-quality SaMD that improves the quality of life for millions of patients.

This opportunity, to improve the quality of life, is why we started Greenlight Guru in the first place. Our MedTech Lifecycle Excellence Platform provides all the tools you need to take your device from design and development all the way through launch and postmarket surveillance. Even better—our world-class medical device Gurus will be with you every step of the way, delivering the expert advice you need, exactly when you need it.

IT'S TIME TO TURN THAT IDEA INTO REALITY.

Get your free demo of Greenlight Guru today →

ULTIMATE GUIDE TO SOFTWARE AS A MEDICAL DEVICE (SAMD)

greenlight guru

317-960-4220

WWW.GREENLIGHT.GURU