

ConnectedHealth

Machine Learning and Medical Devices

Connecting practice to policy *(and back again)*

By Sebastian Holst with Morgan Reed and Brian Scarpelli



Machine Learning and Medical Devices

Connecting practice to policy *(and back again)*

Contents

- Introduction.....2
 - Effective governance is required to accelerate and amplify continued Machine Learning innovation3
 - ML governance must be engineered into ML development practices and account for ML application behaviors3
 - Training Data shapes ML application behavior.....4
 - Source code does not predict ML application behavior4
 - ML applications can continuously evolve.....4
 - Effective governance of ML-enabled solutions begins with effective governance of ML software development and operations5
 - Engineer effective ML governance into Medical Device software development lifecycles5
 - Part 1: Trace ML-specific properties through the software development lifecycle6
 - Part 2: Review Work-In-Progress: A Proposed Regulatory Framework for Modifications to AI/ML-Based Software as a Medical Device6
 - Part 3: Beyond the Total Product Lifecycle7
- Tracing Machine Learning development properties through a general software development and DevOps lifecycle9
 - Software Development Lifecycle Management9
 - Machine Learning SDLC Requirement Summary11
 - Quality Management11
 - ML Software Quality Summary.....13
 - Software Security and Risk Management14
 - Machine Learning Security and Risk Management Summary15
- Work-In-Progress Review: A Proposed Regulatory Framework for Modifications to AI/ML-Based Software as a Medical Device.....16
 - GMLP Summary18
 - The Culture of Quality and Organizational Excellence18
 - Culture of Quality and Organizational Excellence20
- Initial observations21
- Appendix A: Supporting organizations and underlying standards and frameworks22
 - International Electrotechnical Commission (IEC)22
 - International Organization for Standardization (ISO).....22
 - International Medical Device Regulators Forum (IMDRF)23
 - US Food and Drug Administration (FDA)23
- Appendix B: Respondent Submission Analysis24
 - Proposal Questions and Feedback24
 - Respondent industries and corresponding stakeholder community roles25
 - Respondent priorities.....25
 - Respondent priorities by topic26
 - FDA-specific question response.....27
- Appendix C: Beyond the Total Product Lifecycle28

Introduction

Today, there are already many examples of artificial intelligence (AI) systems, powered by streams of data and advanced algorithms, improving healthcare by preventing hospitalizations, reducing complications, decreasing administrative burdens, and improving patient engagement. AI systems offer the promise to further accelerate and scale such results and provide impetus to the ongoing transition from our current disease-based system to one that is centered upon prevention and health maintenance. Nonetheless, AI in healthcare also brings with it a variety of unique considerations for U.S. policymakers, particularly for medical device regulators.

Many organizations are taking steps to proactively address adoption and integration of AI into health care and how it should be approached by clinicians, technologists, patients and consumers, policymakers, and other stakeholders. Building on these important efforts, the Connected Health Initiative's (CHI) Health AI Task Force has taken the next step to address the role of AI in healthcare through the development of health AI policy principles.¹

Generally, CHI believes that AI systems deployed in healthcare must advance the “quadruple aim” by improving population health; improving patient health outcomes and satisfaction; increasing value by lowering overall costs; and improving clinician and healthcare team well-being.

In order to succeed, Health AI systems must:

- Enhance access to health care.
- Empower patients and consumers to manage and optimize their health.
- Facilitate and strengthen the relationship and communication that individuals have with their health care team.
- Reduce administrative and cognitive burdens for patients and their health care team.

In providing its health AI policy principles with various key US federal policymakers, CHI's diverse AI Task Force has identified an opportunity to expand its contribution through a projection of its health AI policy principles onto a collection of good machine learning practices (GMLPs). Through a variety of public and collaborative initiatives designed to refine and build consensus around GMLPs, the objective is to provide a baseline that the Food and Drug Administration (FDA) and other governmental and non-governmental stakeholders can leverage in their ongoing consideration of the topic. We intend for this document to serve as a next step in shaping health AI-related policy developments at the FDA, at the US federal level widely, and internationally.

CHI's AI Task Force welcomes collaboration with any interested stakeholder moving forward and appreciates consideration of this document.

¹ Connected Health Initiative *Policy Principles for Artificial Intelligence in Health*, <https://actonline.org/wp-content/uploads/Policy-Principles-for-AI.pdf>.

Effective governance is required to accelerate and amplify continued Machine Learning innovation

Machine Learning² has advanced the quality and efficiency of medical devices and promises still greater innovations at an ever-quickening pace. Machine Learning's track record coupled with sky-high expectations for the future have also spawned a proportionate demand for – and investment in – effective governance; a means of assessing Machine Learning (ML) application suitability and performance, managing associated risks, and ensuring public safety and ethical use.

This document focuses on governance with respect two primary ML system categories: continuously learning systems (CLS) that are inherently capable of learning from real-world data and are able to update themselves automatically over time while in public use and “locks down” systems that have no ability to alter their configuration once testing and certification have been completed.

Governance strives to ensure appropriate levels of transparency, reliability, safety, security, and privacy.

Effective governance delivers on these objectives without compromising utility, efficiency, or innovation.

Effective ML governance is further required to instill confidence and trust in overall quality that, in turn, will lead to increased development velocity and ever-more ambitious innovation.

ML governance must be engineered into ML development practices and account for ML application behaviors

ML software behaves differently than traditional software in large part because it is developed differently.



The fastest cars need the best brakes. To have the confidence required to drive at the highest speeds, a driver must trust their brakes – not just for emergencies, but for every scenario and under all conditions. And, without exception, the best brakes are engineered into the car; never added on as an afterthought¹.

² CHI supports the exemplary work of numerous organizations that are addressing healthcare AI, and seeks to harmonize and build upon these efforts including reuse, wherever possible, of accepted and recognized terminology and definitions. Unless defined inline, this paper will reuse the terminology and definitions included in the December 2019-released Xavier University paper Building Explainability and Trust for AI in Healthcare. <https://www.xavierhealth.org/news3/2020/1/8>.

³ This analogy has been borrowed with gratitude from the [Open Compliance and Ethics Group](#), a non-profit think tank that promotes Principled Performance as the universal goal of every organization, team and individual.

Training Data shapes ML application behavior



Training Data Set

Rather than explicitly define each logical sequence through source code as a traditional developer would, a ML developer transforms a generic predictive engine (an untrained machine) using a carefully curated training data set. In much the same way that a sculptor creates a mold around an original object, the ML developer creates a trained machine around a training data set. The training data set is constructed by the developer, but the training (computational analysis and resulting modifications to the untrained machine) are executed without developer intervention. The training data set has replaced source code at this stage of the development process and represents a wholly new development artifact.

How should training data sets be created, curated, and vetted?

Source code does not predict ML application behavior



Non-deterministic

There is no longer a one-to-one connection between application logic (behavior) and authored code. Depending on the training data set and the properties of the generic machine selected, the trained engine may have the ability to identify a broken bone in an X-ray, predict a heart attack, or dispense proper dosages of critical medication. Static analysis of peripheral source code or the training data set cannot predict the trained ML engine's behavior.

How can testing criteria be established if software behavior itself cannot be fully specified?

ML applications can continuously evolve



Incr. Learning

Unlike the compilation of source code into an executable program, machine training is not restricted to a single operation prior to an application's production release. If configured to do so, a trained machine that is in production (operational) can employ continuously learning systems (CLS) e.g. continue training using data consumed while in a production environment. This allows for the possibility that different copies of a single trained machine may each evolve independently from one another and from the initial trained machine.

How should new behaviors be evaluated in the field? When can this behavior even be safely deployed?

Effective governance of ML-enabled solutions begins with effective governance of ML software development and operations

The scale, complexity and distribution of ML applications has made governing each ML application instance recommendation, prediction, and action impossible.

What is possible – and practical – is to identify ML-specific risk factors stemming from the “paradigm-shifting” properties outlined above and evaluate how these have been proactively and transparently mitigated *within a broader software development lifecycle management context.*

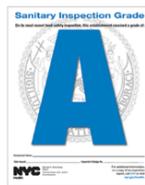


It’s not the “what”, it’s the “how.”

The FDA Food Code ensures food safety and protection by focusing on broad areas of risk including the provisioning, preparation, and delivery of food.

It is not possible to evaluate each of the billions of food servings delivered every day. Governing the food supply chain and preparation “lifecycle” is the only practical means of effective governance.

How to get an A grade in ML software development
“FDA will assess the culture of quality and organizational excellence of a particular company and have reasonable assurance of the high quality of their software development, testing, and performance monitoring of their products².”



Broad Risk Categories	
Food	Machine Learning
Food from unsafe sources	Training data set deficits
Inadequate cooking	Machine training errors
Improper holding temperatures	Pipeline and distribution failures
Contaminated equipment	Operational vulnerabilities
Poor personal hygiene	Poor training and culture

Engineer effective ML governance into Medical Device software development lifecycles

There is an established practice of adapting vetted quality system management and software development lifecycle practices to support the unique priorities and requirements of the medical device industry.

The operative word here is “vetted.” Due in large part to the three paradigm-shifting properties of ML technology outlined above, general ML software quality and development practices may be, in some circumstances, less mature than the development practices currently in place. The potential immaturity of some ML quality and risk management practices suggests that something more than “adapting” generally accepted practices will be necessary.

Given the accrued history and expertise of today’s healthcare software developers – and SaMD developers in particular – this community has a material contribution to make in advancing – not merely adapting – mainstream development best practices.

⁴ Proposed Regulatory Framework for Modifications to Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device (SaMD)

Part 1: Trace ML-specific properties through the software development lifecycle

The first task is to consider where traditional software development and quality management practices are most likely to require ML-specific accommodations prior to suggesting follow-on medical-device-specific adjustments.

The approach taken here is to trace ML-specific properties through the software development lifecycle. In much the same way that a contrast MRI employs a dye to highlight specific and difficult to detect conditions, this paper traces ML-properties across three interwoven software development axes with a special sensitivity to healthcare's overriding priorities, e.g. safety, transparency, and accuracy. The three development axes are:

1. Software manufacturing (the general principles of how whatever is developed is constructed, delivered, and maintained),
2. Software quality management (how suitability of purpose is defined and assessed for what is manufactured), and
3. Software security and risk management (frameworks and practices for identifying, assessing, and mitigating risks stemming from missed manufacturing or quality management requirements).



A Contrast MRI

A contrast MRI uses the injection of a contrast dye to better highlight certain conditions that might otherwise go undetected.

Part 2: Work-In-Progress Review: A Proposed Regulatory Framework for Modifications to AI/ML-Based Software as a Medical Device

In April of 2019, The FDA published an ambitious work that incorporated ML-centric principles into existing software development practices⁵, [Proposed Regulatory Framework for Modifications to AI/ML-Based Software as a Medical Device \(SaMD\) - Discussion Paper and Request for Feedback](#).

The stated goal was to advance a framework that would allow the FDA's regulatory oversight to embrace the iterative improvement power of machine learning for Software as Medical Device while assuring that patient safety is maintained.

Safety assurance is achieved through a multi-pronged approach that includes recommendations that ensure ongoing ML algorithm changes are:

- Implemented according to pre-specified performance objectives,
- Follow defined algorithm change protocols,
- Utilize a validation process that is committed to improving the performance, safety, and effectiveness of AI/ML software, and



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

Discussion Paper and Request for Feedback



⁵ The authors acknowledge their debt to the International Medical Device Regulators Forum (IMDRF) for their work on SaMD (which, itself, relies upon prior IEC and ISO standards and frameworks) while recognizing the need for a “new, total product lifecycle (TPLC) regulatory approach that facilitates a rapid cycle of product improvement and allows these devices to continually improve while providing effective safeguards.”

- Include real-world monitoring of performance.

These recommendations are rolled into an updated Total Product Lifecycle (TPLC) regulatory framework with the ultimate aim of promoting a mechanism for manufacturers to be “continually vigilant in maintaining the safety and effectiveness of their SaMD,” supporting “both FDA and manufacturers in providing increased benefits to patients and providers.”

As with The Food Code, the FDA would assess the culture of quality and organizational excellence of a particular company in order to establish “reasonable assurance” of the high quality of their software development, testing, and performance monitoring of their products.

Given that general-purpose software development practices are themselves undergoing a material ML-driven evolution,

- Are there any underlying assumptions regarding quality and audit that merit closer review?
- What assurances can be built-in to ensure that those changes will be appropriately reflected in the central regulatory notions of “a culture of quality and excellence” and “reasonable assurance?”

Part 3: Beyond the Total Product Lifecycle⁶

Are there untapped approaches to embrace ML’s most dynamic and opaque (but potentially powerful) properties? Are there longer-term opportunities to reimagine certification and pre-certification roles and workflows to further leverage AI/ML innovations?

Perhaps the most radical ML property from a regulatory perspective is the potential for algorithms to evolve after release and distribution. This capability is what is referred to as continuously learning systems.

Currently, this is only a theoretical concern as there is a blanket prohibition of this scenario across every existing and proposed TPLC regulatory framework.

Might there come a time when this prohibition will be perceived as imposing an undue constraint on innovation? Is there a scenario – perhaps in a robotics context – where allowing an initial set of SaMD instances to evolve wholly independently from one another will be identified as an absolute requirement? How would today’s notions of manufacturing lifecycle and quality need to adapt?

The FDA, Machine Learning & SaMD

The FDA’s has already begun the complex task of reimagining regulatory oversight to best embrace the power of machine learning while continuing to assure patient safety.



Only “frozen algorithms” need apply (for now)

As with a graduating class of identically trained physicians whose skills mature independently over time, it is possible for an initial set of ML SaMD instances to evolve wholly independently from one another after distribution.

Might there come a time when the prohibition of real-time, continuous learning is perceived as an undue constraint on innovation?

⁶ See Appendix C: Beyond the Total Product Lifecycle

Machine Learning is not the only transformative computing force. Cloud services, mobile 5G, and blockchain are among a growing list of revolutionary technological domains that are enabling entirely new ways of working, collaborating, and communicating.

Are there near-term organizational or technological opportunities that can help to prioritize near-term ML regulatory, governance and compliance requirements while also better positioning stakeholders across the healthcare and technology spectrum to capitalize on what may appear at first to be ML's most radical properties?

Tracing Machine Learning development properties through a general software development and DevOps lifecycle

Healthcare software governance combines policies and controls to:

- Ensure public safety
- Mitigate risks stemming from
 - Unintended consequences
 - Poor execution
 - Adversarial exploitation
- Encourage innovation in applications as well as the specialized development and testing tools required to produce those applications.

In what ways might ML development properties challenge foundational assumptions underlying traditional development lifecycle management practices?

Software Development Lifecycle Management

Software Development Lifecycle (SDLC) Management and DevOps tooling and practices normalize and automate software manufacturing processes while helping to ensure that safety, transparency, and privacy requirements are met.

In order for Machine Learning to complete its transition from paradigm-shifting innovation to a mainstream technology, SDLC management must also meet any additional requirements stemming from ML data-driven machine training development practices, e.g. Machine Learning Software Development Lifecycle Management (MLDLC).

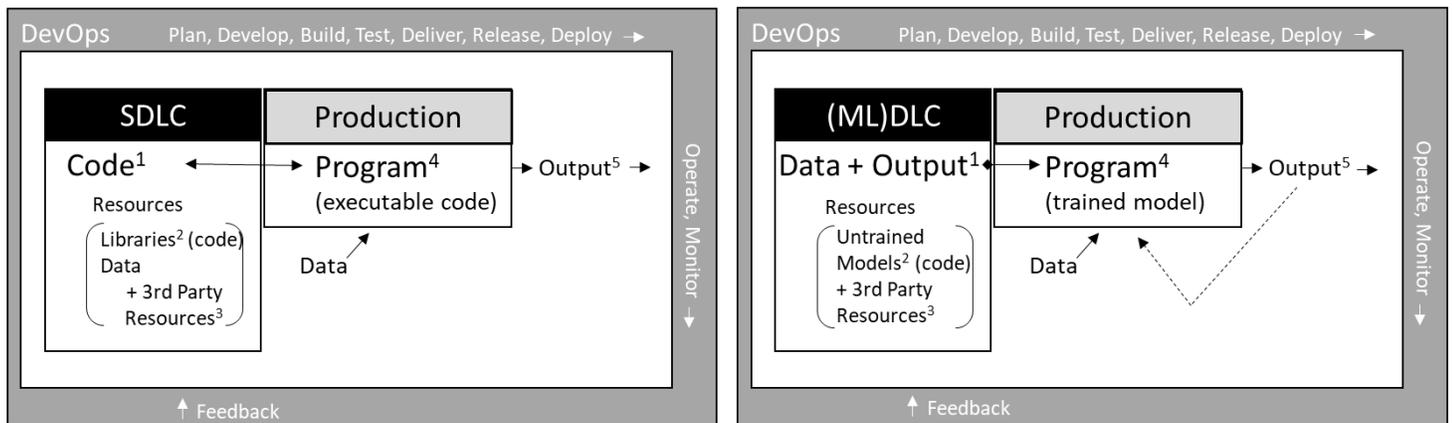


Figure 1: Traditional SDLC versus Machine Learning MLDLC wrapping in a DevOps iterative pipeline.

Figure 1 illustrates the elements of, and relationships between, a traditional Software Development Lifecycle and a Machine Learning Development Lifecycle operating within a well-formed DevOps pipeline.

The Figure 1 notes are described in the following table.

ML Key	Topic	Note
	1 Code vs Data + Output	Code sits at the center of a traditional SDLC and, consequently, is subject to rigorous quality, audit, and sourcing controls. Given that Data + Output supplants Code in a ML DLC, it follows that <i>an equivalent – but not identical – collection of controls are needed to ensure that effective quality, audit and sourcing remain in place.</i>
	2 Libraries vs Untrained Models	A traditional SDLC has built-in support for managing reusable code, typically in the form of libraries, to speed and simplify development, improve quality and auditability, and to help ensure consistency over time and across development teams. In much the same fashion, ML DLC will draw from a collection of reusable untrained models ⁷ . These models are code-based and are often organized as a traditional library, but given their heightened impact on development outcomes, <i>a corresponding increase in Untrained Model governance may also be justified.</i>
	3 3 rd Party Resources	Today’s applications increasingly rely upon 3 rd party managed services, libraries, and software components. SDLC tools (Integrated Development Environments or IDE’s) as well as software and service distribution channels have been extended to better support this rapidly evolving software supply chain. Supply chain risk management has also evolved to ensure appropriate visibility and accountability as the sourcing of code and services become increasingly distributed and diverse. <i>IDE’s and IT security and risk management frameworks must evolve in-kind to keep pace with the consequences of including 3rd party Data + Output and/or Untrained Models into the modern software supply chain.</i>
	4 Production Programs	The traditional SDLC deliverable is an executable program. The ML DLC deliverable is a trained model. Due to ML statistical techniques, it is typically not possible – or nearly impossible – to trace exactly why a trained model behaves as it does. The absence of a decision tree in an ML program renders traditional SDLC code reviews, debugging, and general monitoring techniques obsolete. <i>ML programs may require compensating mechanisms to ensure comparable degrees of transparency, reliability and auditability.</i>
	5 Output	Both traditional SDLC and DevOps best practices include a feedback loop that can be used to generate new requirements or improve existing features. This kind of continuous feedback fuels future program iterations and is subjected to the complete SDLC beginning with requirements through coding, test, etc. However, there are some branches of Machine Learning, specifically Continuously Learning where feedback is delivered directly into the current Production ML Program. These classes of Machine Learning bypass traditional SDLC inspection and approval steps and may result in unplanned and, potentially, unexpected behaviors. <i>Owners and regulators of sensitive and high-risk applications that must include human inspection may need to consider a blanket prohibition of these subcategories of Machine Learning until new norms about acceptable risk and transparency can be established. At a minimum, a greater understanding of the limitations and side-effects of deployed machine learning algorithms will be required by auditors and regulators.</i>

Table 1: ML DLC requirements stress traditional SDLC practices.

⁷ ML programs also include “traditional reusable code” as well.

Machine Learning SDLC Requirement Summary

Tracing ML properties through high level SDLC stages suggested several potential new or modified requirements including:

1. The transition from code-driven to data-driven development will require corresponding practices and controls to meet quality, audit, and sourcing requirements.
2. Reusable Untrained Models are a special class of reusable code that, given their heightened impact on development outcomes, require a proportionate increase in governance.
3. Security and risk management must evolve in-step to keep pace with the implications of including 3rd party Data + Output and/or Untrained Models into the modern software supply chain.
4. Production ML programs may require novel monitoring and debugging mechanisms to ensure acceptable transparency, reliability, and auditability
5. Owners and regulators of sensitive and high-risk applications may need to consider blanket prohibitions of CLS Machine Learning models unless and until revised notions of transparency and predictability are established.
6. Integrated Development Environments (IDE's) and associated tooling will need to be extended to better scale and automate all phases of the new MLDC.

Quality Management

While SDLC management measures and manages software manufacturing, distribution, and consumption, Software Quality is the field of study and practice that describes, measures, and manages the desirability (suitability) of the software itself.

Production Software Quality is, in large part, built upon Software Program Quality (the executable) that is, in turn, built upon the underlying Code Quality.

The shift to trained models away from code suggests a requirement to supplement existing code-centric quality practices and metrics.

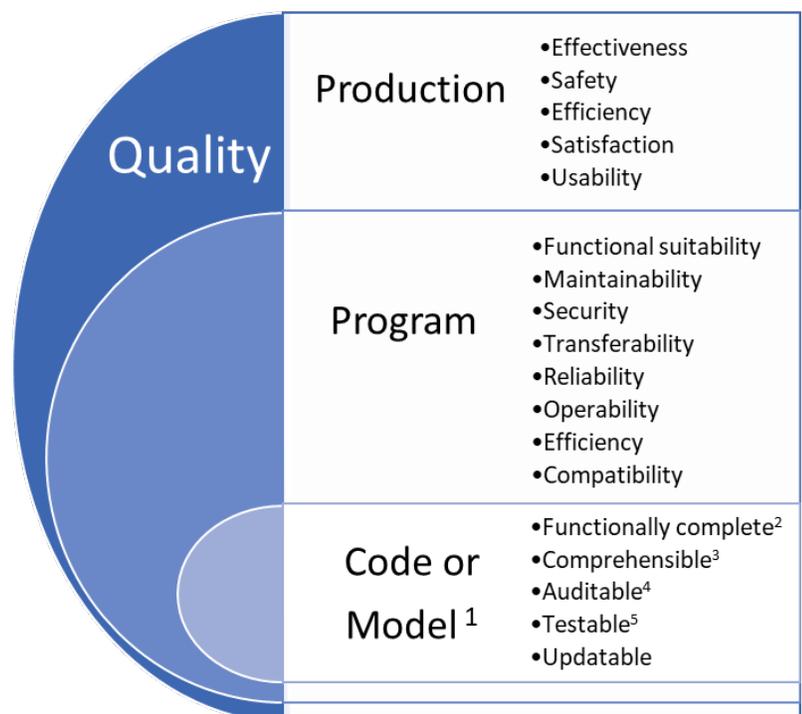


Figure 2: Quality is managed throughout the development lifecycle.

Figure 2 illustrates the elements of, and relationships between, common quality metrics divided into three segments: underlying code (or trained model), the resulting program, and the performance or suitability of that program.

Figure 2 notes are described in the following table:

ML Key	Quality Topic	Note
	1 Code vs Trained Model	Code sits at the center of a traditional Software Quality Practice with well-defined subcategories including functional completeness, comprehensibility, auditability, testability, and updatability. To preserve overall Quality, <i>ML development must develop equivalent – but not identical – methods of measuring and establishing acceptable quality metrics and tolerances.</i>
Trained Model vs Code		
	2 Functionally Complete	<p>Code can be statically analyzed, monitored for “coverage”, and otherwise exercised to generate a mapping of input data and environmental states to expected outcomes.</p> <p>ML models are trained and tested through the processing of carefully curated data sets – there is no code that can be parsed and traced. Poorly formed datasets generate unexpected and potentially unpredictable, behaviors and/or incorrect weighting of outcome predictions. Common examples of training data set gaps include:</p> <ul style="list-style-type: none"> - Insufficient data volume - Lopsided data distribution across activities and outcomes - Missing activities and/or outcomes - Impossible activities or outcomes <p>Poor data sets can result in the compromise multiple functional subcategories including:</p> <ul style="list-style-type: none"> - Suitability: will the software behave appropriately for all users? - Accuracy: are functions implemented correctly? The models themselves may meet the highest quality standards, but the resulting trained model may fail to meet those standards. - Compliance: is the software in compliance with the necessary laws and guidelines? Transparency and predictability are required with virtually every regulatory and/or compliance obligation. <p><i>Development must have reliable means of detecting and, as needed, remediating gaps and other data set irregularities prior to ML model training.</i></p>
	3 Comprehensible	<p>Every ML model includes intrinsic limitations. Understanding the stated purpose and objectives of a ML application and the hosting platform and implementation language will not be sufficient to assess the suitability of either training data or the selected ML models. In order to meaningfully “comprehend” the expected behavior of a trained model, <i>a reviewer must have specialized data science expertise and be knowledgeable in the strengths and limitations of the applied model(s) and the data staging/cleansing/sampling techniques.</i></p>
	4 Auditable	<p>Tracing, reverse-engineering, and predicting how a model will behave given a specific set of inputs is difficult and, in practical terms, often impossible. This is especially true with extremely complex systems with many thousands of variables; the most common examples include image recognition, robotics, and natural language processing. <i>A consensus on acceptable alternatives to traditional event logging in code-based applications are needed to provide a comparable degree of assurance.</i></p> <p>Untrained models are often provided by open source communities or platform providers. <i>A common format for sourcing the precise model and version with a record</i></p>



		<i>of know Quality issues would help to predict Quality issues that may arise in the final trained model.</i>
5	Testable	<p>Exception detection, defect definition, and related KPI's (including testing cost) must be established to effectively model the severity and cost of ML application defects specifically related to under-performance.</p> <p>Output measurement must also be standardized, utilizing what developers measure for their own data models including terminology and their own interpretation of medical information. <i>This industry-specific formulation results in a harmonization of terminology across regulators and stakeholders that will improve quality management.</i></p>

Table 2: Trained Models drive expansion of code-centric Software Quality practices.

ML Software Quality Summary

Tracing ML properties across basic Quality System segments suggested several additional new or modified requirements including:

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. ML Software must meet the same quality standards as code-based software. As such, there must be equivalent methods of measuring and establishing acceptable ML-centric quality metrics and tolerances to offset inapplicable code-centric controls. 2. ML-centric controls must cover both the special data sets used for training and testing ML models as well as the trained ML models themselves. 3. Reviewers, testers, and auditors will require additional specialized data science expertise including a working knowledge of the strengths and limitations of deployed model(s), the implications of their parameters as well as any data staging/cleansing/sampling techniques that are applied. 4. The sourcing of untrained models is a potential supply chain gap – in much | <ol style="list-style-type: none"> 5. Quality Systems must also incorporate updated and harmonized health care specific terminology, data collection, and measurement practices to ensure the availability of relevant baseline healthcare quality metrics and standards. 6. The establishment of exception detection, defect definition, and related KPI's (including testing cost estimation) are needed to effectively model the severity and cost of ML application defects specifically related to ML under-performance. |
|--|---|

Software Security and Risk Management

Effective risk and security management begins with identifying and prioritizing material threats and works to establish effective controls that reduce risk to acceptable levels. For application risk and security management, recommended practices typically include:

- Detailed Abuse Cases⁸ that are used to
 - Develop a business/technical specific Threat Model⁹ that in turn is used to assess risks stemming from
 - Each application's Attack Surface¹⁰, e.g. the application's entry and exit points.

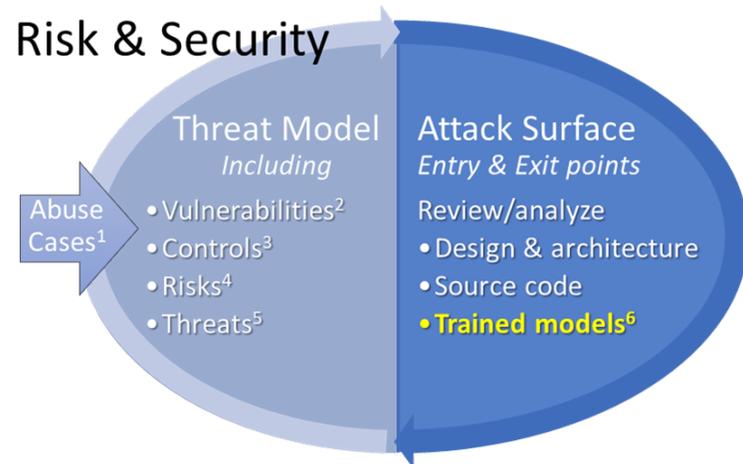


Figure 3: Risk and Security Modeling

These interrelated components evolve with production usage and feedback generating additional Abuse Cases that in turn update the Threat Model resulting in further refinements to the application's Attack Surface and underlying controls.

Software Security and Risk Management practices must also expand to meet new requirements stemming from Machine Learning development practices, technology, and use cases. Figure 3 notes are described in the following table.

ML Key	Risk & Security	Note
	1 Abuse Cases	<i>The current paucity of established ML Abuse Cases is likely to lead to an incomplete view of potential threats and undermine threat modeling activities and the subsequent control priorities that follow.</i>
	2 Vulnerabilities	ML systems novel use of training data to create production behaviors have spawned an equally novel set of novel vulnerabilities including: <ul style="list-style-type: none"> • Data poisoning (injecting training data designed to cause errors) • Adversarial input (data crafted to be misclassified by targeted models) • Exploitation of errors in autonomous system goals <i>The set of known ML-specific vulnerabilities is almost certainly incomplete as are the range of potential exploits.</i>
	3 Controls	<i>There is a further deficit in established Preventative and Detective Controls to mitigate the risks stemming from ML-inspired vulnerability attacks.</i>
	4 Risks	Effective risk assessments are dependent upon accurate probability estimates. Risk calculations typically combine:

⁸ OWASP Abuse Case Cheat Sheet

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Abuse_Case_Cheat_Sheet.md

⁹ OWASP Application Threat Modeling

https://www.owasp.org/index.php/Application_Threat_Modeling#1._What_are_we_building.3F

¹⁰ OWASP Attack Surface Cheat Sheet

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Attack_Surface_Analysis_Cheat_Sheet.md



		<ul style="list-style-type: none"> • The probability of an incident occurring (an exploit of a vulnerability) • The probability of that incident causing harm and • The degree of harm that comes with each occurrence <p><i>The rapidly evolving use of ML across industries and use cases significantly complicate ML risk assessment calculations making risk mitigation investment decisions more difficult to calibrate.</i></p>
5	Threats	<p>In addition to the exploitation of unique ML vulnerabilities, the weaponization of ML in the hands of bad actors must also be considered. Examples include:</p> <ul style="list-style-type: none"> • Automation of social-engineering attacks and the dissemination of political misinformation leveraging improved profiling, messaging and deep fake image and audio generation. • Anonymization and scaling of physical assaults using autonomous drones and other vehicles • Highly efficient and distributed cyber-attacks leveraging specialized ML models. • Expansion of potential attackers as democratization of all of the above removes human domain expertise as a requirement. <p><i>ML expands the variety of potential threats, improves the efficiency of existing threats, and expands the number of potential attackers.</i></p>
6	Trained models	<p><i>ML training and test data sets represent additional attack surface opportunities to be included in current Attack Surface mapping practices.</i></p>



Table 3: Machine Learning impact on established Application Risk and Security practices

Machine Learning Security and Risk Management Summary

Tracing ML properties through security and risk management categories highlight some measure of risk from all three ML property categories listed above.

1. The short history of successful ML exploits constrains Threat Modeling practices.	making risk mitigation investment decisions more difficult to calibrate.
2. The inventory of ML-specific vulnerabilities is incomplete as are the understanding of potential exploits.	5. ML training and test data sets represent additional attack surface opportunities to be included in current Attack Surface mapping practices.
3. There is a further deficit in established Preventative and Detective Controls to mitigate the risks stemming from ML-inspired vulnerability attacks.	6. ML has a multiplicative effect on Risk and Security management by expanding the variety of potential threats, improving the efficiency of existing threat tactics, and expanding the number of potential attackers
4. The rapidly evolving use of ML across industries and use cases significantly complicate ML risk assessment calculations	

Work-In-Progress Review: A Proposed Regulatory Framework for Modifications to AI/ML-Based Software as a Medical Device

[A Proposed Regulatory Framework for Modifications to AI/ML-Based Software as a Medical Device \(SaMD\) - Discussion Paper and Request for Feedback](#) was published with the stated goal of advancing a framework to allow the FDA’s regulatory oversight to embrace the iterative improvement power of machine learning for Software as Medical Device while assuring that patient safety is maintained.

The proposed Total Product Lifecycle (TPLC) regulatory framework is designed to ensure ongoing ML algorithm changes are:

- Implemented according to pre-specified performance objectives,
- Follow defined algorithm change protocols,
- Utilize a validation process that is committed to improving the performance, safety, and effectiveness of AI/ML software, and
- Include real-world monitoring of performance.

In order to manage the scale and scope of this ambitious effort and to avoid the necessity of auditing every development milestone of every software component, the FDA proposes assessing the culture of quality and organizational excellence of a particular company in order to establish “reasonable assurance” of the high quality of their software development, testing, and performance monitoring of their products.

As outlined in the prior section, much of the underlying general-purpose software development standards, frameworks, and practices¹¹ are themselves actively undergoing their own ML-driven evolution. This section drills into the updated Total Product Lifecycle Regulatory approach and the associated “Culture of Quality and Organizational Excellence” to identify:

- Underlying assumptions regarding Software Development Lifecycle Management, Quality or Risk that may merit closer review, and
- Mechanisms to ensure evolving assumptions are appropriately reflected in the central notions of “a culture of quality and excellence” and “reasonable assurance.”

In order to “balance the benefits and risks, and provide access to safe and effective AI/ML-based SaMD,” the revised TPLC seeks to establish clear expectations on quality systems and good ML practices (GMLP) as outlined in the following illustration.

¹¹ See Appendix A: Supporting organizations and underlying standards and frameworks.

GMLP Summary

Evaluating GMLP in the context of the ongoing evolution of ML-centered development quality, SDLC, and risk management, the following issues may merit deeper investigation:

1. Heavy reliance on standards that have historically been defined by methodical and deliberate revision policies may not be able to keep pace with rapidly changing development practices and exacerbate rather than mitigate quality risk stemming from ML's data-driven versus code-driven properties.
2. Without a sufficient body of verified ML development patterns have been documented, it may be difficult to establish a durable definition of "reasonable" and "effective."
3. The long-standing requirement that all copies of a given device or software instance can only be updated but cannot independently evolve prohibits a subset of dynamic and continuously learning applications.
4. Incident management and platform monitoring systems will likely need to expand incident categories and severity ratings to account for unique classes of exceptions unique to ML services.

The Culture of Quality and Organizational Excellence

The Culture of Quality and Organizational Excellence is itself comprised of three management principles:

1. Leadership that sets the organizational tone,
2. Lifecycle Support Processes that wrap and operationalize the actual development, and at its core,
3. Deployment, and maintenance activities associated with actual SaMD development.

As noted in Table 4, note 1 above, software lifecycle standards, such as IEC 62304, are code centric and will likely need to be extended or adapted to the unique lifecycle requirements associated with training ML algorithmic models.



FDA SaMD QMS Principles

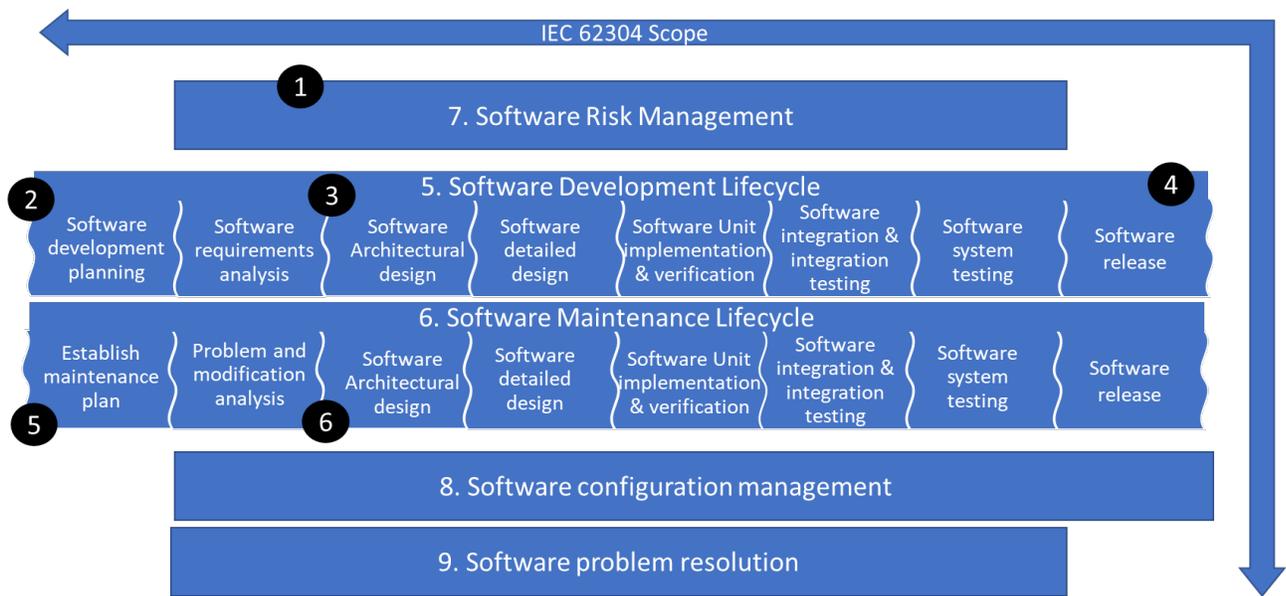


Figure 5: ML development impact on IEC 62304 development lifecycle processes.

Figure 5 notes are described in the following table.



Note	ML development impact on IEC 62394 development lifecycle processes.
1	Software Risk Management: <i>How will risks associated with training data sets be mitigated?</i>
2	Software development planning: <i>How will Software Of Unknown Providence (SOUP) be extended to accommodate 3rd party algorithms and external training data?</i>
3	Software requirements analysis: <i>How will issues relating to bias and transparency be incorporated?</i>
4	Software release: <i>Given the requirements above, how can FDA Premarket Safety Assurance requirements be effectively be met?</i>
5	Maintenance plan: <i>Defining, measuring, and documenting the degree of change within an SaMD will require significant coordination and consensus.</i>
6	Problem and modification analysis: <i>Documenting root causes and effectivity of modifications stemming from data set deficiencies will require new (or enhanced) concepts, tooling and terminology.</i>

Table 5: ML development considerations within IEC 62304: Medical device software lifecycle processes.

Culture of Quality and Organizational Excellence

Evaluating working definition of the Culture of Quality and Organizational Excellence in the context of the ongoing evolution of ML-centered development quality, SDLC, and risk management, the following issues may merit deeper investigation:

1. To satisfy an external auditor/examiner, Organizations will need to be able to tap into a sufficiently large body of recognized ML controls able to substantially meet their requirements.
2. Suppliers of third party and embedded software, also referred to as Software Of Unknown Provenance (SOUP), must be able to satisfy corresponding requirements for transparency, safety, security, and privacy.
3. Individuals will need the ability to know if/how their data may be used to develop and/or train machines or algorithms. The opportunity to participate in data collection for these purposes must be on an opt-in basis.^{12 13}
4. A consensus must be reached on the definition and measurement of a wholly new quality criteria related to behavior, e.g. bias and human-readable decision-making transparency.
5. New (or enhanced) concepts, tooling and terminology will likely be required across a broad spectrum of operations management capabilities to properly capture the impact of dataset deficiencies including:
 - Chance control documentation including risks assessment,
 - Root cause analysis, and
 - Modification effectiveness.

¹² Connected Health Initiative *Policy Principles for Artificial Intelligence in Health*, <https://actonline.org/wp-content/uploads/Policy-Principles-for-AI.pdf>.

¹³ American Medical Association's privacy principles <https://www.ama-assn.org/system/files/2020-05/privacy-principles.pdf>.

Initial observations

There is wide agreement that existing regulations need revision to accommodate the unique (and potentially disruptive) properties of Machine Learning technologies and development processes.

100% of Proposed Regulatory Framework responses endorsed the requirement to update existing medical device regulatory obligations to accommodate Machine Learning¹⁴.

The FDA, responding to this need, has proposed a regulatory framework to manage what is likely to be one of the most challenging aspects of regulating ML-driven “Software as Medical Devices,” modifications that may, or may not, require a review and recertification – a potentially time-consuming and expensive process.

One of the distinguishing properties of the Machine Learning approach is the capacity for programs to alter behavior over time without requiring additional coding or software updates. This kind of unsupervised learning challenges conventional development, quality, and risk practices and policies.

The FDA proposal built off existing regulations, frameworks, and definitions, extended some where needed, and added wholly new constructs when it was determined to be unavoidable.

Initial feedback to the proposed framework reinforced the importance of leveraging existing standards and framework – perhaps to an even greater extent than the initial proposal envisioned.

There is significantly more work that needs to be done refining and harmonizing definitions, completing core processes and performance metrics, as well as educating the vast community of stakeholders.

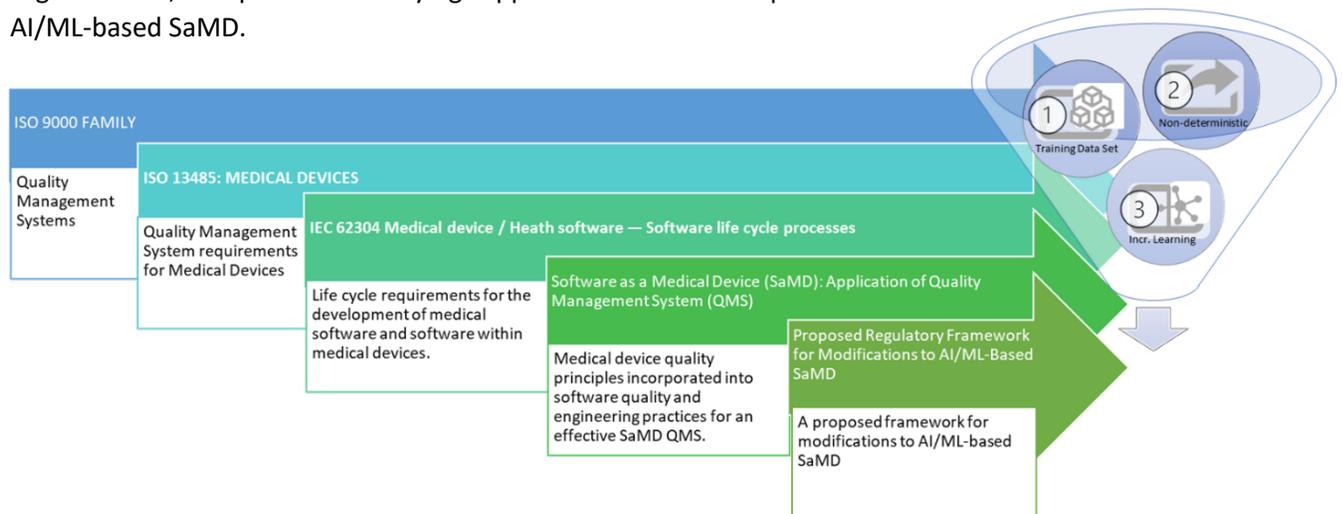
Tracing ML-specific development and technical properties from Innovator practices through relevant tooling, development frameworks, and standards promises to ultimately shorten and simplify the work required to effectively and efficiently “protecting the public health by ensuring Software as Medical Device safety, efficacy, and security.”

This can be most effectively accomplished through a sustained collaboration with, and communication across, the stakeholder ecosystem (innovators, platform providers, supranational standards bodies, government regulators, etc.).

¹⁴ See Appendix B: Respondent Submission Analysis

Appendix A: Supporting organizations and underlying standards and frameworks

There is an established practice of adapting vetted quality system management and software development lifecycle practices to support the unique priorities and requirements of the medical device industry. The following list includes frameworks and documents, as well as the associated governing organizations, that provide underlying support for the FDA's Proposed Framework for Modifications to AI/ML-based SaMD.



International Electrotechnical Commission (IEC)

The IEC prepares and publishes International Standards for all electrical, electronic and related technologies.

[IEC 62304:2006/AMD 1:2015](#) Medical device software life cycle processes is a standard which specifies life cycle requirements for the development of medical software and software within medical devices.

International Organization for Standardization (ISO)

ISO is an independent, non-governmental international organization with a membership of 164 national standards bodies. ISO – in conjunction with the IEC – has identified the need to develop standards for AI that “can benefit all societies.” Established in 2017, this is the charter of the ISO/IEC Joint Technology Committee (JTC) 1 / Subcommittee (SC) 42 for artificial intelligence (SC 42).

SC 42’s scope includes basic terminology and definitions, risk management, bias and trustworthiness in AI systems, robustness of neural networks, machine-learning systems and an overview of ethical and societal concerns. SC 42 has already published three Big Data standards with 13 projects currently under development. Five of these are highlighted below.

[ISO/IEC JTC 1/SC 42](#): Artificial Intelligence

AI/ML ISO standards under development from ISO/IEC JTC 1/SC 42 include:	
ISO/IEC 23053	Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)
ISO/IEC 24027	Bias in AI systems and AI aided decision making
ISO/IEC 38507	Governance implications of the use of artificial intelligence by organizations
ISO/IEC 23894	Artificial Intelligence — Risk Management
ISO/IEC TR 24368	Artificial Intelligence (AI) — Overview of ethical and societal concerns

[International Medical Device Regulators Forum \(IMDRF\)](#)

The IMDRF is a voluntary group of medical device regulators from around the world who have come together to form the Global Harmonization Task Force on Medical Devices (GHTF) whose mission is to “accelerate international medical device regulatory harmonization and convergence.” Their relevant works to date are highlighted here.

IMDRF publications include:	
IMDRF/SaMD WG/N10	SaMD: Key Definitions
IMDRF/SaMD WG/N12	SaMD: Possible Framework for Risk Categorization & Corresponding Considerations
IMDRF/SaMD WG/N23	SaMD: Application of Quality Management System
IMDRF/SaMD WG/N41	SaMD: Clinical Evaluation

[US Food and Drug Administration \(FDA\)](#)

The FDA is responsible for protecting the public health by ensuring the safety, efficacy, and security of drugs, biological products, *and medical devices*. In addition to the [Proposed Regulatory Framework for Modifications to AI/ML-Based Software as a Medical Device \(SaMD\) - Discussion Paper and Request for Feedback](#), the FDA is also active in contributing to, endorsing, and re-publishing many of the IMDRF publications listed above. At this time, the FDA has not made ML-specific modifications to Medical Device regulatory obligations (see [21 CFR Parts 803 through 861](#)).

Appendix B: Respondent Submission Analysis

Proposal Questions and Feedback

While there were no constraints placed on the kinds of feedback or questions that could be submitted, the FDA included questions that covered the most important (or perhaps controversial) elements of the proposed TPLC framework.

Questions included in Proposed Regulatory Framework were divided into subtopics.

- How complete is the classification of AI/ML SaMD modifications and will they be effective and helpful?
- Is the GMLP complete? How can the FDA help manufactures incorporate new requirements into their existing QMS systems and practices?
- All feedback to the definitions and implementation details surrounding SPS and ACP. These are entirely new elements to the proposed certification process.
- How can the process of premarket review (review prior to an initial SaMD launch) be better defined and managed?
- How can “real-world” data be captured, analyzed, secured, and weighted throughout this entire process?
- What should the ACP include and how can it be consistently and effectively assessed across manufacturers and SaMDs?

These questions bring to the fore just how potentially disruptive Machine Learning may be in the short-term – and why it is in everyone’s interest to shorten the ML transition into the mainstream.

That being the case, why did 64% of respondents fail to answer even one of the FDA’s questions?

64% of the public responses did not directly reference a single question included in the Framework Proposal.

Questions included in Proposed Regulatory Framework

The types of AI/ML-SaMD modifications (Key: **AI/ML SaMD**)

1. Do these categories of AI/ML-SaMD modifications align with the modifications that would typically be encountered in software development that could require premarket submission?
2. What additional categories, if any, of AI/ML-SaMD modifications should be considered in this proposed approach?
3. Would the proposed framework for addressing modifications and modification types assist the development AI/ML software?

Good Machine Learning Practices (Key: **GMLP**)

1. What additional considerations exist for GMLP?
2. How can FDA support development of GMLP?
3. How do manufacturers and software developers incorporate GMLP in their organization?

SPS and ACP (Key: **SPS/ACT**)

1. What are the appropriate elements for the SPS?
2. What are the appropriate elements for the ACP to support the SPS?
3. What potential formats do you suggest for appropriately describing a SPS and an ACP in the premarket review submission or application?

Premarket review (Key: **PreMarket**)

1. How should FDA handle changes outside of the “agreed upon SPS and ACP”?
2. What additional mechanisms could achieve a “focused review” of an SPS and ACP?
3. What content should be included in a “focused review”?

The transparency and real-world performance monitoring

(Key: **Transp & Monitoring**)

1. In what ways can a manufacturer demonstrate transparency about AI/ML-SaMD algorithm updates, performance improvements, or labeling changes, to name a few?
2. What role can real-world evidence play in supporting transparency for AI/ML-SaMD?
3. What additional mechanisms exist for real-world performance monitoring of AI/ML-SaMD?
4. What additional mechanisms might be needed for real-world performance monitoring of AI/ML-SaMD?

ACP Scope: (Key: **ACP**)

1. Are there additional components for inclusion in the ACP that should be specified?
2. What additional level of detail would you add for the described components of an ACP?

The following analysis is based upon the public responses to The Proposed Regulatory Framework for Modifications to AI/ML-Based Software as a Medical Device (SaMD) - Discussion Paper and Request for Feedback.

Looking at the respondents' own questions and/or their interest (and/or lack of interest) in the FDA's questions offers insight into how stakeholders outside of the FDA perceive these issues and which of these may be perceived as more (or less) important or controversial.

Respondent industries and corresponding stakeholder community roles

Respondent submissions are available for review on the FDA website¹⁵. Figure B1 maps the self-identified Industry Categories of 127 respondents to generic Stakeholder Community roles¹⁶.

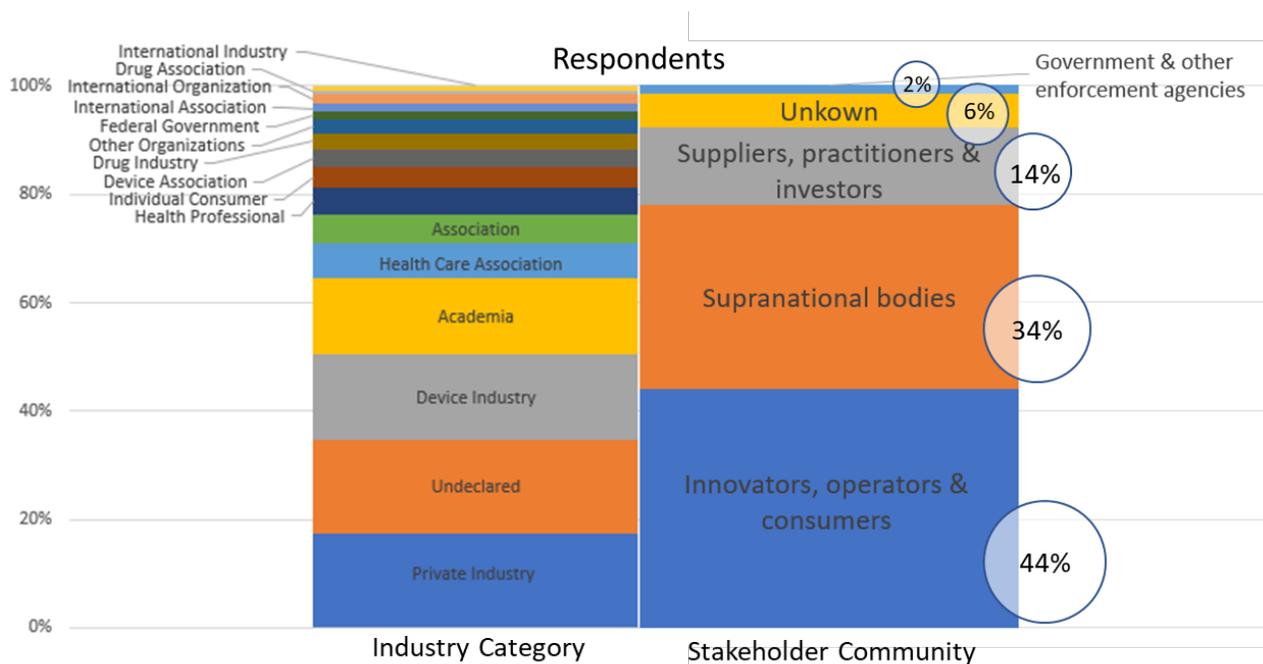


Figure B1: Respondent Industry Categories and Stakeholder Roles

Perhaps it is not surprising to learn that the primary stakeholders have the loudest voice (at least by sheer volume), but, given the importance of vendor-neutral, independent “Supranational bodies” in shaping regulations, should they?

Respondent priorities

The questions embedded inside the FDA's regulatory framework proposal are calibrated to address the FDA's priorities, but are those priorities and their relative weighting shared? Figure B2 illustrates the percentage of responses that included specific topics. These topics are grouped into “framework-specific” (that are unique to the proposed regulatory framework) and “mainstream activities” (that are general issues already described relating to the mainstreaming of any disruptive technology).

¹⁵ <https://www.regulations.gov/document?D=FDA-2019-N-1185-0001>

¹⁶ When included, the respondent's organization was also used to map into the Stakeholder Community role.

64% of respondents did not answer any of the 18 questions included in the proposal. Closer inspection of respondents' comments suggests a difference in emphasis and, perhaps, priority.

Respondents that did answer FDA-specific questions:

1. Were much more likely to comment on the ML SaMD modification categories, the recertification criteria and process, and the description of the TPLC.
2. Consistently raised issues across the mainstream activities of Quality, Risk, Ecosystem (collaboration across roles) and Frameworks (reconciliation with other frameworks).
3. Respondents that did not answer the FDA-specific questions were significantly more likely to focus on software Quality and Risk issues.
4. Regardless of whether the FDA-specific questions were addressed, there was a general concern around the definition and treatment of "Locked" models.

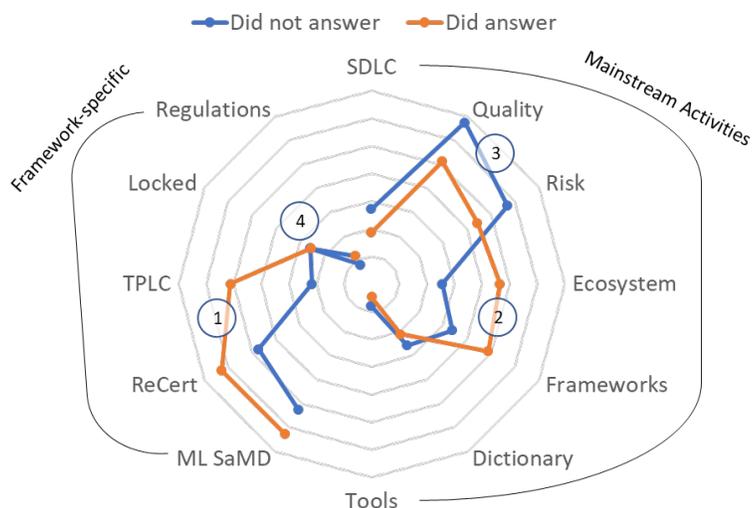


Figure B2: Topic interest of respondents (collaboration across roles) and Frameworks (reconciliation with other frameworks).

Respondent priorities by topic

Does a respondent's stakeholder role as innovator or standards body (versus regulatory agency or consumer) also influence their priorities? If yes, should the dominance of one stakeholder role over all others be factored-in or weighted when considering responses?

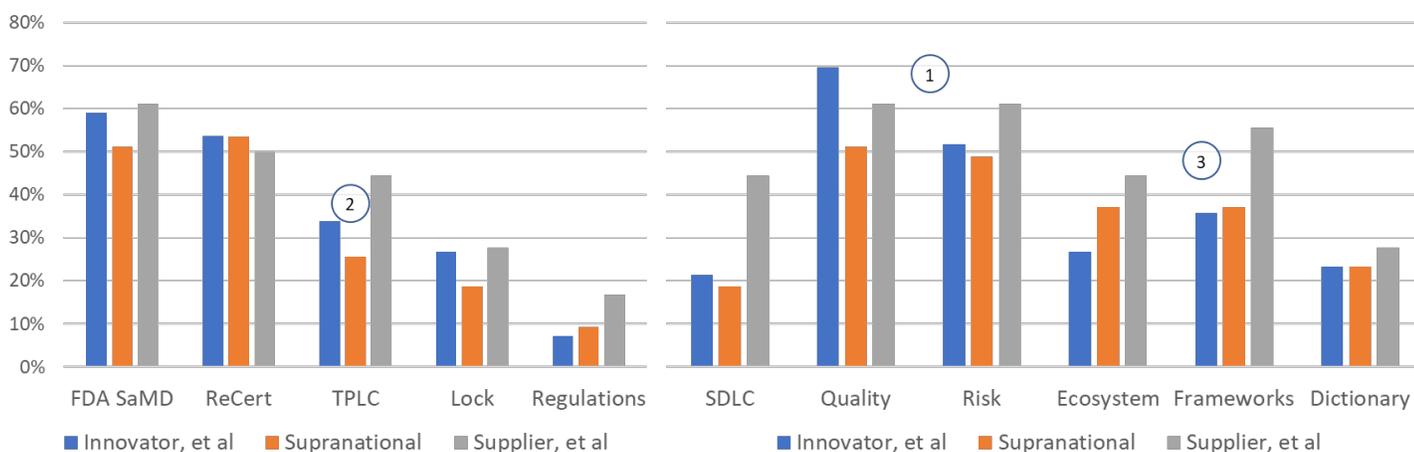


Figure B3: percentage of responses across topics by Ecosystem Stakeholder role.

Figure B3 maps the percentage of topics included in responses by Stakeholder role (only three roles had enough responses to be statistically meaningful).

1. Quality, Risk, FDA SaMD modifications and recertification processes received the greatest attention.
2. Generally, Innovators, consumers, practitioners and suppliers responded more consistently with one another as compared to Supranational organization responses.

3. Taken as a group, comments relating to Ecosystem (cross roll collaboration), Frameworks (cross framework reconciliation), and Dictionary (defining common terms and definitions across domains) were a strong, consistent area of concern.

FDA-specific question response

While only 36% of respondents addressed the embedded 18 questions directly, those responses were extensive and, obviously, important to assess.

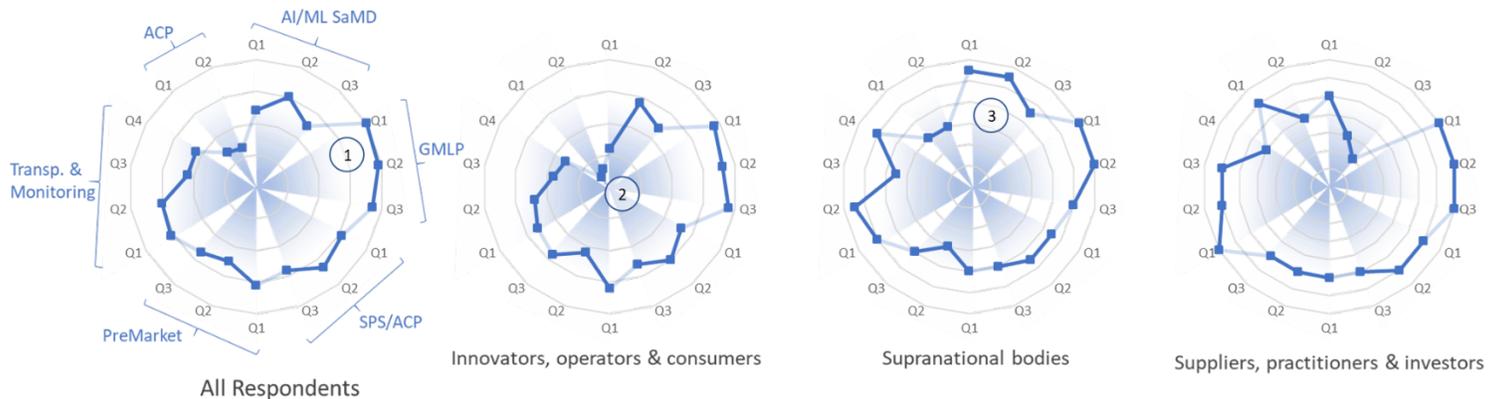
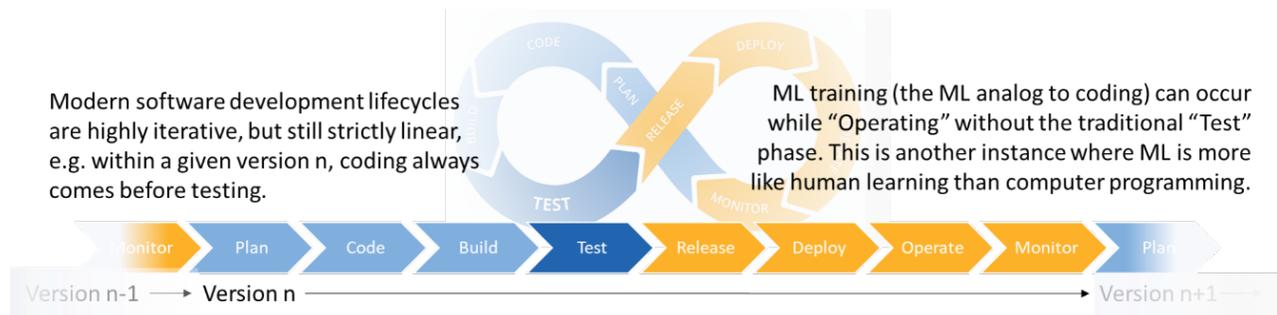


Figure B4: Count of responses that included commentary for each FDA-embedded question. The questions are segmented by topic. All Respondents are shown alongside the three highest reporting Ecosystem Stakeholder roles.

1. Respondents gave the greatest amount of attention to the questions relating to Good Machine Learning Practices.
2. Relative to the other subtopics, Algorithm Change Protocol received substantially less attention from Innovators, et al than any other subtopic. This gap was not evident in either of the other two Stakeholder roles.
3. The high innovator response volume depressed the relative importance of the ACP subtopic. Given the close relationship between Supranational Organizations and Government Regulators already discussed and the consensus around the importance of framework and regulatory consistency, should the (apparent) lack of interest from Innovators be discounted?

Appendix C: Beyond the Total Product Lifecycle

Software development lifecycle management, like virtually all modern Product Lifecycle Management, is a highly iterative process, but within any given version, the lifecycle stages are executed in a strictly linear sequence. As an example, within a given version n, coding, building, and testing must always



precede deployment and production operation.

When configured to do so, continuously learning algorithms can breach the strict sequencing imposed by development lifecycle methodology. Not surprisingly, the FDA’s proposed AI/ML TPLC includes a prohibition of this kind of evolutionary behavior in real-time and in production. This is a sound policy as there is no precedent to contradict this position to be found in the underlying standards and frameworks.



Yet, while there is no *underlying* precedent, might there be a precedent to be found in an *adjacent health care domain*?

Who’s Who and What Do They Do?

To assure patient safety, every healthcare worker must, on a reoccurring basis, be credentialed by an array of professional, State and Federal agencies.

Expensive and time consuming: Credentialing costs the U.S. healthcare system billions of dollars per year and it is time consuming. Credentialing one physician takes, on average, 100 days; a time period where that physician cannot practice.

Thanks to encrypted digital ledgers, mobile technology, and cloud services, this seemingly intractable bureaucratic nightmare is being reimaged and rebuilt as a high-speed, on-demand service able to support existing regulatory and statutory obligations at scale – improving patient safety and increasing healthcare professional availability.

If this technology can be trusted to credential hundreds of thousands of mobile healthcare professionals – what would it take to credential and authenticate millions of continuously learning medical devices?

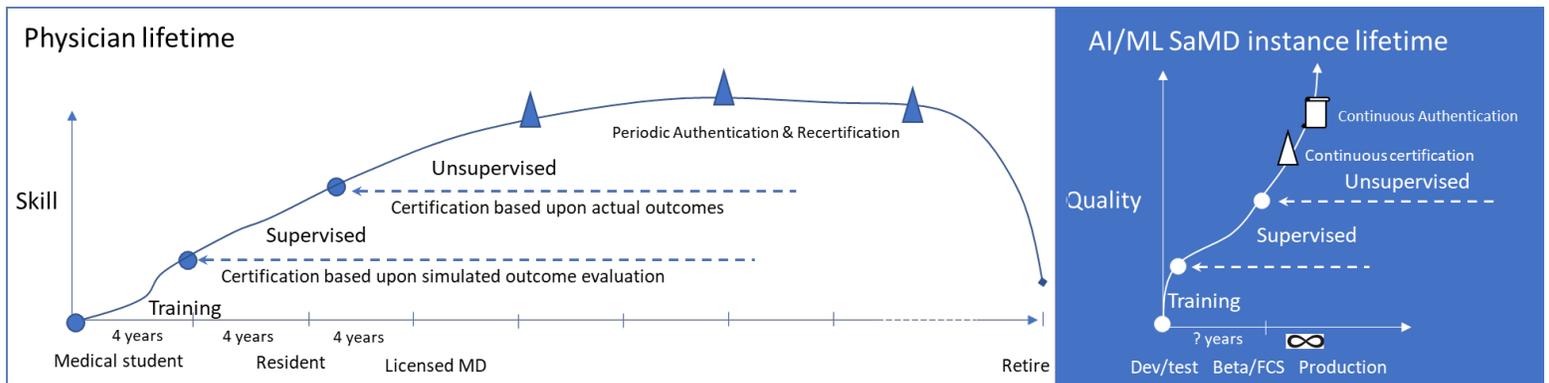


Figure C1: modeling an individual SaMD instance Quality as an independent healthcare worker's Skill.

The training, testing, and certification of a physician is not unlike the (ML)DLC or the FDA's GMLP for an AI/ML SaMD. The two only truly diverge after "certification." A physician is expected to continue to learn and improve – often in ways that are distinct from other physicians who were part of the same graduating class, a.k.a. the same release.

While there are governing bodies and controls in place to monitor the maturation of each individual physician – and to remove their privileges when needed – AI/ML SaMDs cannot be monitored individually today. As such, to assure patient safety, individual SaMD instance continued growth cannot be permitted.

Could a similar technology cocktail of encrypted digital ledgers (blockchain), mobile, and cloud technologies scale to reliably authenticate and then certify each individual medical device instance?

The first question that needs to be asked and answered is what innovation or benefits will be lost if continuous learning in production cannot be deployed. If there is no compelling use case, subsequent issues around monitoring and regulating their safety are moot.

What is evident is that, in order to remain relevant and support innovation, every interested party must remain open to reimagining the traditional roles and relationships between innovators, regulators, patients, service providers, et al alongside the coming waves of ML discoveries and breakthroughs.