Proposals for AI Use in Medical Device Software Development

Gerald Rigdon
Fellow, Software Engineering
Boston Scientific Corporation
St. Paul, USA
gerald.rigdon@bsci.com
May 2026

Abstract

Since the Medical Device Amendments were added to the Federal Drug and Cosmetic Act in 1976, the FDA has been charged with the oversight of an industry with substantial and ever-emerging technological change over the past 50 years. Medical devices have evolved from mechanical devices to sophisticated electro-mechanical products operated by complex software. With the advent of connected medical devices in recent years, the FDA has had to deal with a new era where devices are now subject to cybersecurity breaches. And now on the heels of responses to pressing security concerns, the agency faces the expansive frontier of AI. This paper offers an MDM perspective and identifies important proposals for the use of AI in medical device software development.

Key Terms:

AI – Artificial Intelligence
DUS – Direct Use Software
FDA – Food and Drug Administration
LLM – Large Language Model
MDM – Medical Device Manufacturer
ML – Machine Learning
SME – Subject Matter Expert
SUS – Supporting Use Software

Introduction

In January 2024, the United States Government Accountability Office (GAO) published GAO-24-106122 [1] as part of a collaborative effort by several federal agencies to consider the regulatory opportunities and challenges posed by emerging technologies. These findings were presented as a report to congressional requestors regarding the use of Artificial Intelligence (AI) as related specifically to medical devices under the authority of the Food and Drug Administration (FDA). As discussed in a GAO highlights and summary document [2], the purpose of the broader GAO investigation was to examine:

(1) challenges and opportunities that government agencies report facing in regulating emerging technologies

(2) government agency collaboration and cooperation activities
(3) lessons that government agencies can learn from other governments' experiences

As a result of these investigative goals, the GAO report [1] found the FDA deficient given the agency had not yet presented any well documented information to Congress on the use of AI in medical devices. This was a concern since Congress was expected to move ahead without any specific direction on necessary legislative changes which could translate into missed opportunities to "fully realize the public health benefits of this technology." Further it recommended the following action for the FDA: "The Commissioner of FDA should identify and document the specific changes to its statutory authorities that would enable FDA to take the actions it determines best to oversee AI/ML-enabled medical devices and then communicate these potential legislative changes to Congress."

FDA Analysis and Discussion

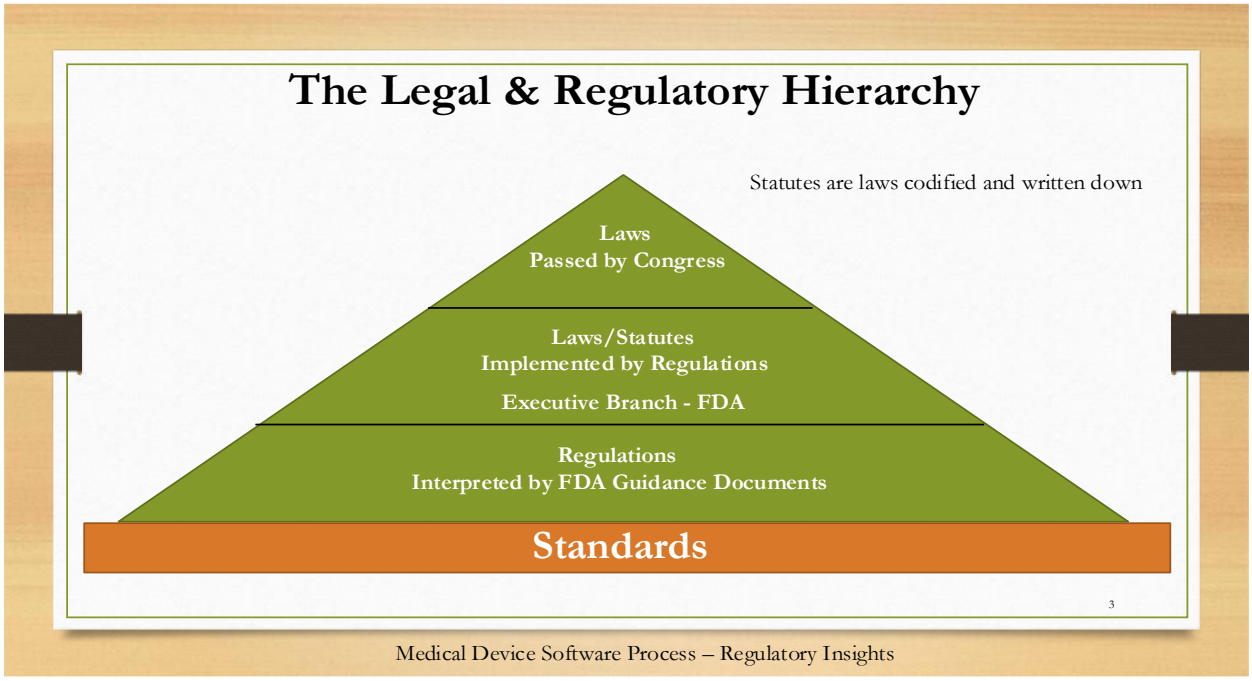

Figure 1

As depicted in Figure 1, the FDA is an arm of the U.S. Executive Branch of government and has no authority to create laws. Instead, laws are passed by Congress and then implemented by the Agency (FDA) in the form of regulations known as the Code of Federal Regulations. Additionally, the regulations are further interpreted by the FDA in the form of guidance documents and typically state what is found in the 2025 Cybersecurity Guidance [3]: "In general, FDA's guidance documents do not establish legally enforceable responsibilities. Instead, guidances describe the Agency's current thinking on a topic and should be viewed only as recommendations…" This is pertinent given the GAO report [1] recommended actively seeking input from the FDA that would be used to create binding law. However, in doing so, the GAO report did acknowledge the downsides of regulating too quickly which could impede progress and so focused on 5 U.S.C.

Section 553 which describes formal and informal rule making, the latter being the recommended process to be followed.

In defense, it should be noted that the FDA had not been completely silent on the AI topic. Between 2019 and 2021 there were discussion papers on AI/ML learning which included proposed frameworks and action plans. Further, in 2021 the FDA in cooperation with Health Canada and the U.K.'s Medicines and Healthcare Products Regulatory Agency identified ten guiding principles [4] for machine learning. In October 2023, the FDA released a bulletin [5] citing 171 AI/ML enabled devices that were added to their existing list of such devices and stated: "This list is not meant to be an exhaustive or comprehensive resource of AI/ML-enabled medical devices. Rather, it is a list of AI/ML-enabled devices across medical disciplines, based on publicly available information. The FDA maintains this list to increase transparency about AI/ML-enabled devices in this rapidly progressing field." Following GAO report [1], FDA has published other documents including guiding principles for machine learning transparency [6] in June 2024 and the software focused marketing submission final guidance for AI enabled devices [7] in August 2025.

Medical Device Software in Focus

With this historical backdrop and context in place, attention is now turned to the focus of this paper. In contrast to the deployment of AI models and/or algorithms instantiated in medical device products that provide patient therapy, attention will now be given to the use of AI in the development of software in said devices.

With respect to this broader topic the January 2025 International Medical Device Regulators Forum (IMDRF) recommendations [10], referenced by the FDA, aims to provide direction for AI/ML use in medical device development with what is described as "guiding principles." Although these principles are propounded in a document discussing development, the content presents more generally. While one could extract some guidance for development, the emphasis of the material is managing what is deployed in products. More pertinent is the FDA's own January 2025 draft guidance [11] on AI-enabled device software functions. This material revolves around AI deployed in devices, but the content addresses the Total Product Life Cycle (TPLC) which includes design and development activities. However, these activities are identified as supporting "the design, development, deployment, and maintenance of AI-enabled devices." For clarity, the draft guidance defines an AI-enabled device as follows:

"AI-enabled devices are devices that include one or more AI-enabled device software functions (AI-DSFs). An AI-DSF is a device software function that implements one or more 'AI models' (referred to as 'models' in this guidance) to achieve its intended purpose. A model is a mathematical construct that generates an inference or prediction based on new input data. In this guidance, when 'AI-enabled device' is used, it refers to the whole device, whereas when 'AI159 DSF' is used, it refers only to the function that uses AI."

For the purpose of this paper, understanding these definitions and drawing out necessary distinctions is crucial to the proposals presented herein and answering questions that arise, namely:

What is the guidance for the use of AI in the development of medical device software for any device?

What are the expectations governing the use of AI in implementing software, reviewing software, producing documentation, generating tests, verifying test results, etc., of Software in a Medical Device (SiMD) or Software as a Medical Device (SaMD)?

Given medical device constraints, particularly SiMD used in embedded systems, one could argue that in the near-term the most prevalent use cases in such devices will be in the role of AI in development of software, not the instantiation of AI in software. Helpful to this discussion is the FDA validation guidance [12] from 2002, where the third bullet captures the essential point being made:

- Software used as a component, part, or accessory of a medical device
- Software that is itself a medical device (e.g., blood establishment software)
- *Software used in the production of a device (e.g., programmable logic controllers in manufacturing equipment)*
- Software used in implementation of the device manufacturer's quality system (e.g., software that records and maintains the device history record)

Framed in a different way, the question becomes: Is AI used in manufacturing, design and development, or other parts of the quality system?

Further insight in answering this question is presented in the 2026 FDA software assurance guidance [13]. This guidance starts with intended use and then follows with a risk-based approach. Compliance with its core message is presented in the following steps:

Step 1: Is it DUS or SUS?

With respect to intended use, per [13] the software subject to 21 CFR 820.70(i) includes the following categories:

- Category 1: Direct Use Software (DUS)
  - Software intended for automating production processes, inspection, testing, or the collection and processing of production data.
  - Software intended for automating quality management system processes, collection, and processing of quality management system data, or maintaining a quality record established under applicable quality management system obligations.

- Category 2: Supporting Use Software (SUS)
  - Software intended for use as development tools that test or monitor software systems or that automate testing activities for the software used as part of production or the quality management system, such as those used for developing and running scripts or software embedded in the production equipment (e.g., firmware).

- Software intended for automating general record-keeping for production or the quality management system that is not part of the quality record.

These are the categories that are required to have software validation per 21 CFR 820.70(i). Overall, the software assurance guidance [13] embraces a risk-based approach while remaining steadfast to the necessity of validation and identifying what is subject to it. Although AI is not specified in the DUS and SUS sections of [13], it is identified in the overarching framework which states: "The approach outlined can be applied, but is not limited, to automation tools (e.g., BOTS or automatic workflows), data analytic tools, artificial intelligence/machine learning tools, and cloud computing when used as part of production or the quality system."

Step 2: Is it High Process Risk or Not High Process Risk?

Once the intended use is identified, the risk-based approach is applied. Process risk is simplified into two categories. Presented here is an interpretive summary:

- High process risk – This is where a failure of the software feature, function, or operation could directly impact production or the quality system in a way that also creates medical device risk. In these cases, [13] requires assurance activities that are commensurate with the device risk, meaning more thorough, rigorous, and documented verification and validation activities.

- Not high process risk – While these situations still require assurance activities, the rigor and extent are proportional to the actual process risk. Even if an MDM internally calls the risk "moderate" or "low," it still falls into this "not high process risk" category for the purposes of the guidance [13].

This binary classification is used so that MDMs focus their efforts and documentation on the features, functions, and operations that pose the greatest risk to product quality and patient safety. The approach applies not only to traditional automation tools but also to data analytics, AI/ML, BOTS, cloud computing (when used in production/quality systems), and similar technologies within scope**.**

Given AI/ML is not the focus of [13], Figure 2 is an attempt to synthesize the content and the various examples from the guidance and illustrate the use of AI in this context.

Figure 2

Step 3: What are the other quality assurance activities?

The software tooling and infrastructure components are largely governed by 21 CFR 820.70, especially clause (i), which refers to the broader reaching "quality system." Yet much of the focus for those who develop medical device software is Design Controls from 21 CFR 820.30. As of 2026 [14], the transition has begun to amend the entire medical device QSR part 820 to align with the 2016 version of ISO 13485 [15].

The challenge for MDMs is understanding how to apply the correct level of software quality assurance activities (rigorous vs streamlined). To address how an MDM can go about mapping the software quality assurance content in [13], one of the key questions presented in the outset should be revisited:

What is the guidance for the use of AI in the development of medical device software for any device?

Pursuant to the answer, the following adaptation principles are suggested.

Key Adaptation Principles:

- Replace process risk with device feature risk.
- Tie the risk determination to potential impact on patient safety and device effectiveness, not production quality.
- Align assurance activities with software development lifecycle controls — testing, verification, validation — applicable to device features.

Introduction to Proposals 1 & 2 – Staying Upstream

First, the following important definitions from [12] are revisited:

"Software verification provides objective evidence that the design outputs of a particular phase of the software development life cycle meet all of the specified requirements for that phase. Software verification looks for consistency, completeness, and correctness of the software and its supporting documentation, as it is being developed, and provides support for a subsequent conclusion that software is validated."

"Software validation is a part of the design validation for a finished device, but is not separately defined in the Quality System regulation. For purposes of this guidance, FDA considers software validation to be 'confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled.'"

While determining DUS vs SUS and High Process Risk vs Not High Process Risk are left to each MDM to resolve, good arguments could be made for each combination. For example, it is compelling to argue that AI (which is software) when used to develop other software (even medical device software) is SUS, and with a few minor tweaks to SUS description language in the guidance [13] it would be unambiguous. However, such straightforward wording is absent, and whether intentional would require FDA input. Therefore, this paper attempts to defend **Proposal 1** and **Proposal 2** by assuming the most conservative approach namely, High Process Risk — DUS.

In general, satisfying validation is a multi-step process that involves creating a validation plan, determining the intended use represented by requirements, constructing a validation protocol and test specification, and then finally conducting tests and documenting results. This paper contends the following proposals do not fundamentally alter the medical device software development process and the only noticeably different input is the use of AI (or more precisely LLMs) in the upstream software development process which occurs before the downstream validation items.

This is captured Figure 3, where the same software verification and validation procedures and processes continue to operate as usual regardless of the upstream activities that precede them.

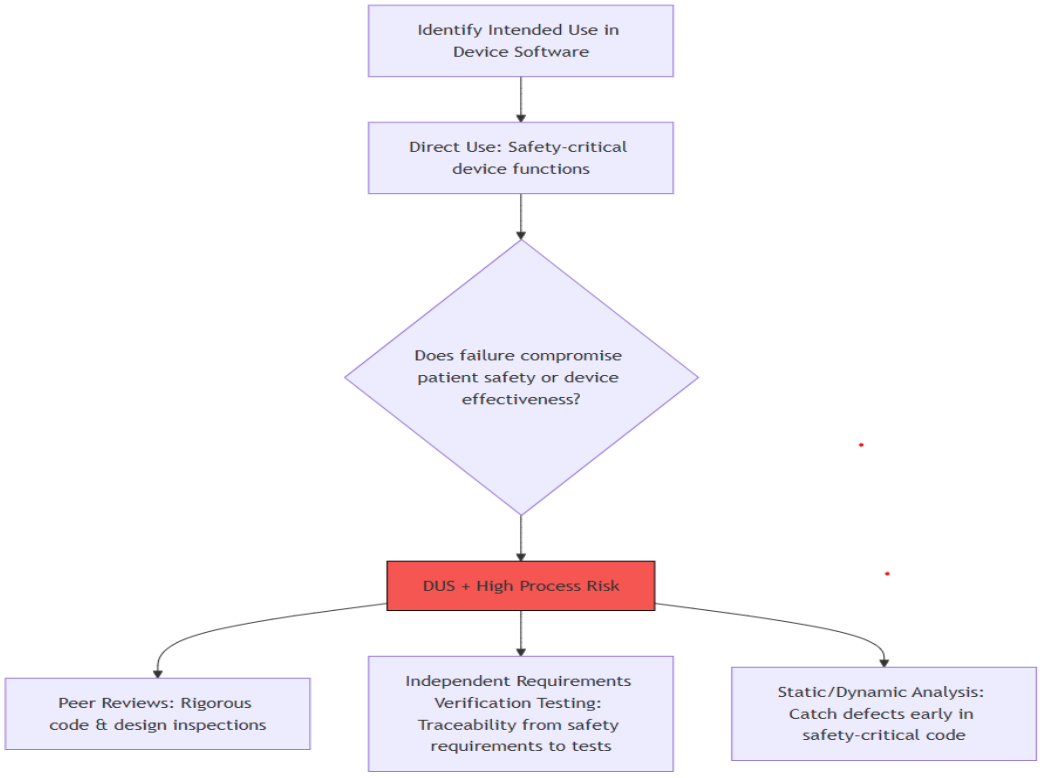

Figure 3

Code Generation Proposal

Proposal 1: Use of LLMs for Generation of Medical Device Code

Code generation may be the most obvious use case though not necessarily the least controversial. Pushback against this use case though is difficult to defend given the acceptance of downstream validation and the assumption that it is relied upon in this proposal. Given a set of requirements, there are virtually endless ways to implement the solution.

Hence, a crucial question arises. Assuming the implementation meets peer review criteria (including all associated quality, maintenance, and performance requirements, etc.), is coding standard compliant per static analysis tools, and passes all tests, does the origin of the implementation matter? Often the implementation of a particular software feature is assigned to an available and hopefully engaged human resource. Given the arbitrary nature of this selection process, does it matter if software engineer X is an LLM?

Consider the following software implementation process proposal presented in Figure 4.

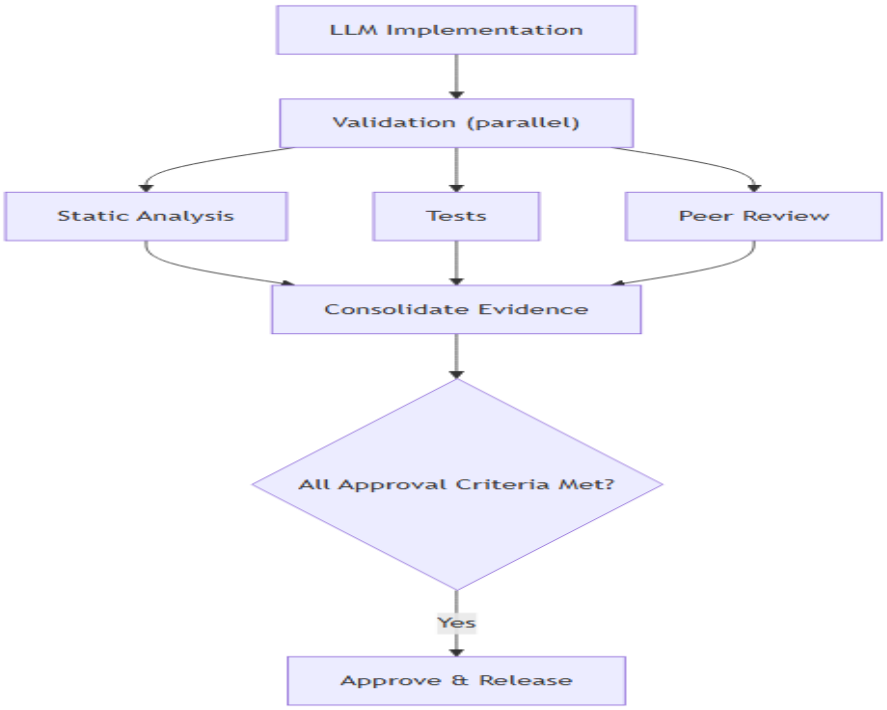

Figure 4

In this process proposal, the LLM takes on the role of software implementation. The upstream LLM role is validated by success in meeting all the downstream approval criteria.

As a whole, this proposal:

- Satisfies all requirements
- Has the potential to significantly reduce coding development time
- Frees a human engineer to pursue other tasks best suited for business interests

Unit Test Proposal

Proposal 2: Use of LLMs for Unit Test Generation

Proposing LLM usage for generating tests must be weighed carefully given **Proposal 1** relies on the testing process for downstream validation. However, it must be appreciated that testing is multi-faceted and this proposal narrows focus on unit testing only. Therefore, **Proposal 1** remains defensible given grey-box Integration Testing and black-box Verification Testing processes remain conventional without any LLM involvement.

Revisiting some of the concerns addressed in **Proposal 1**, any kind of code or test script generation is non-deterministic. In other words, if one invokes the exact unit test generation sequences back-to-back using the same LLM, it is possible to get different results. In the case of code generation, the concerns were alleviated by the demonstration of downstream validation activities. For unit testing, downstream validation rests on the conventional unit test tooling framework. This tooling enforces the same test coverage and test file syntax requirements regardless of the test script author identity. Hence, concerns over variations in test script generation should be put in perspective given the vast variation of test approaches that can be taken depending on the human engineer assigned to the task.

Figure 5 captures the process using a unit test tool.

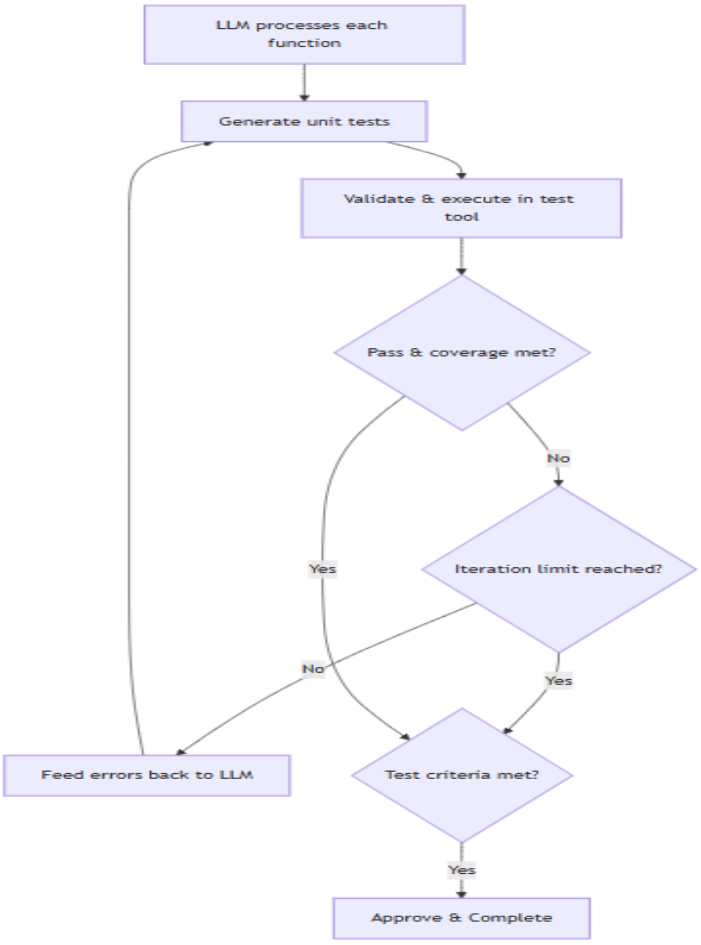

Figure 5

In this process proposal, the LLM takes on the role of unit test script generator. The upstream LLM role is validated in a downstream reflection loop that provides any unit test tool syntax error feedback to subsequent LLM input context until the unit test tool framework is satisfied and finalized by an approval process.

As a whole, this proposal:

- Satisfies all unit test tool coverage and test file syntax requirements
- Has the potential to significantly reduce unit test development time
- Frees a human engineer to pursue other tasks best suited for business interests

Introduction to Proposal 3 – Going Further Downstream

Having defended **Proposal 1** and **Proposal 2**, attention is now given to further thought experiments considering the potential of LLMs to be used in downstream activities. In this proposal, an LLM is used in only one arm of the peer review process, namely the independent reviewer. The subject matter expert remains and anchors a human in the loop.

Peer Review Proposal

Proposal 3: Use of LLMs in Peer Review Process – LLM as an Independent Reviewer

Per the formative 2002 validation guidance [12] that is still cited in recent FDA documents, it states:

"Validation activities should be conducted using the basic quality assurance precept of 'independence of review.' Self-validation is extremely difficult. When possible, an independent evaluation is always better, especially for higher risk applications. Some firms contract out for third-party independent verification and validation, but this solution may not always be feasible. Another approach is to assign internal staff members that are not involved in a particular design or its implementation, but who have sufficient knowledge to evaluate the project and conduct the verification and validation activities. Smaller firms may need to be creative in how tasks are organized and assigned in order to maintain internal independence of review."

This raises at least two important questions. Can LLMs be used as independent peer reviewers? If so, how can they be validated for this task?

Software tools are presently ubiquitous in design, implementation, and test verification processes. Framed in this manner as a tool, one might concede that peer review is a possible use case, assuming validation for intended use like every other tool occurs. Revisiting [12], the recommendation is "to assign internal staff members that are not involved in a particular design or its implementation, but who have sufficient knowledge to evaluate the project and conduct the verification and validation activities." With respect to the assessment of such a reviewer, the FDA guidance is not prescriptive, deferring instead to a risk-based judgment on selecting a competent reviewer that was not involved in the design or implementation.

Thus: Is an LLM trained by a vast corpus of software engineering principles and coding languages not qualified for such a task unless validated specifically for this task? What would this type of

validation prove in this context? Does one validate the independent human peer reviewer in this way?

For software related peer reviews, perhaps thinking more about the use of static analysis tools can be helpful to understanding and defending this proposal. Static analysis tools became popular and recommended by the FDA because they excel in vetting software in ways more suitable to computers and less suitable for the human brain. Would one prefer a human to evaluate a repository of complex C code for MISRA C compliance or a static analysis tool executed by a computer?

Consider the following software review process proposal presented in Figure 6.

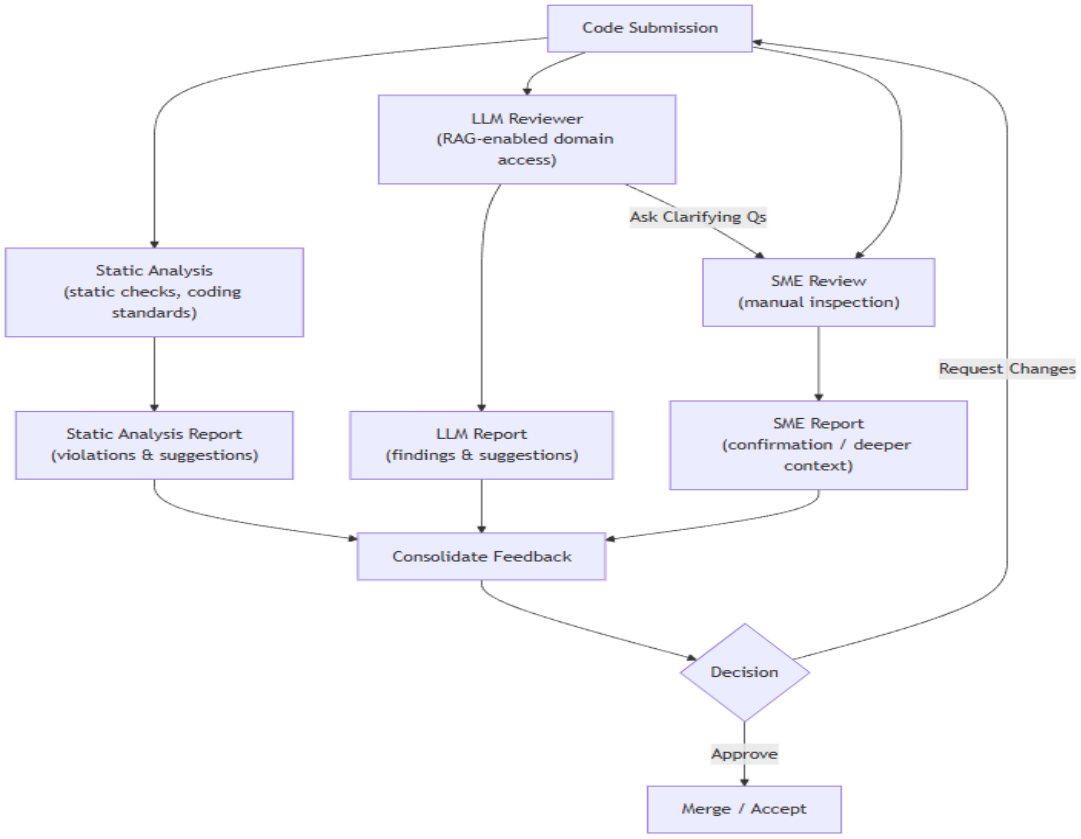

Figure 6

In this process proposal, the LLM can be thought of as another type of static analysis tool. As a complement to conventional static analysis tools enforcing coding standards, the LLM using Retrieval Augmented Generated (RAG) leverages domain related data (requirements, design documentation, design constraints, etc.) in addition to its base training to produce a unique analysis. As a whole, this proposal:

- Keeps the human Subject Matter Expert (SME) and coding standard compliance in the loop
- Arguably, provides a more effective review
- Frees a human independent reviewer to pursue other tasks best suited for business interests

Proposal 1 (code generation) relies on the peer review process for downstream validation but the use of an LLM has been introduced in **Proposal 3**. However, as discussed, peer review remains anchored with an independent SME and coding standard compliance inputs. For those not satisfied with this rationale, consider a further option in making the present proposal viable:

- one could evaluate validity of an LLM as a reviewer by a separate human reviewed process

Summary

This paper provides a regulatory and technical exploration of how AI, particularly LLMs, can be applied to the development of medical device software, distinct from AI-enabled devices. Against this backdrop, three proposals are presented: (1) using LLMs to generate medical device code, (2) employing LLMs to produce unit tests, refined through automated tooling loops, and (3) deploying LLMs for independent peer review, complementing static analysis and SME oversight. The paper emphasizes that these uses of LLMs can improve efficiency and reduce resource strain while staying compliant through documented processes, targeted internal checks, and risk-based validation, and it calls for continued industry and regulatory dialogue as AI adoption accelerates.

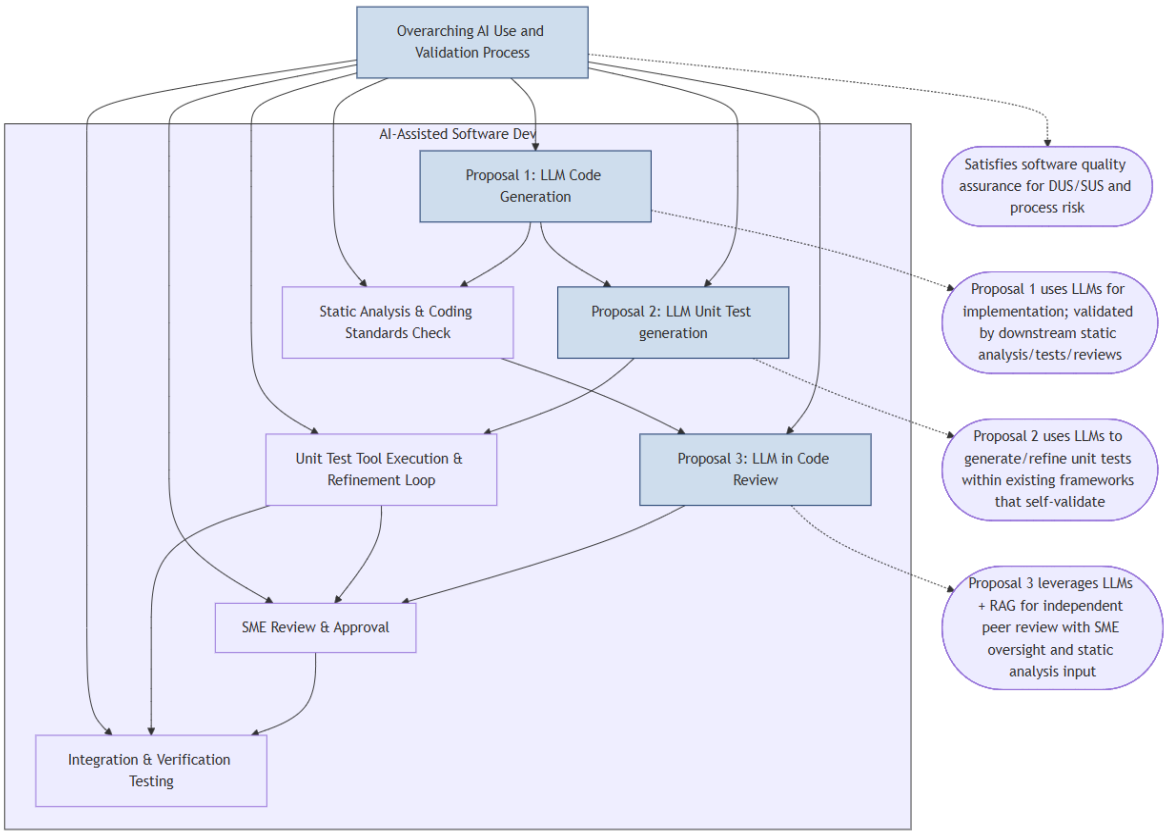

Figure 7

Risk-Based Analysis

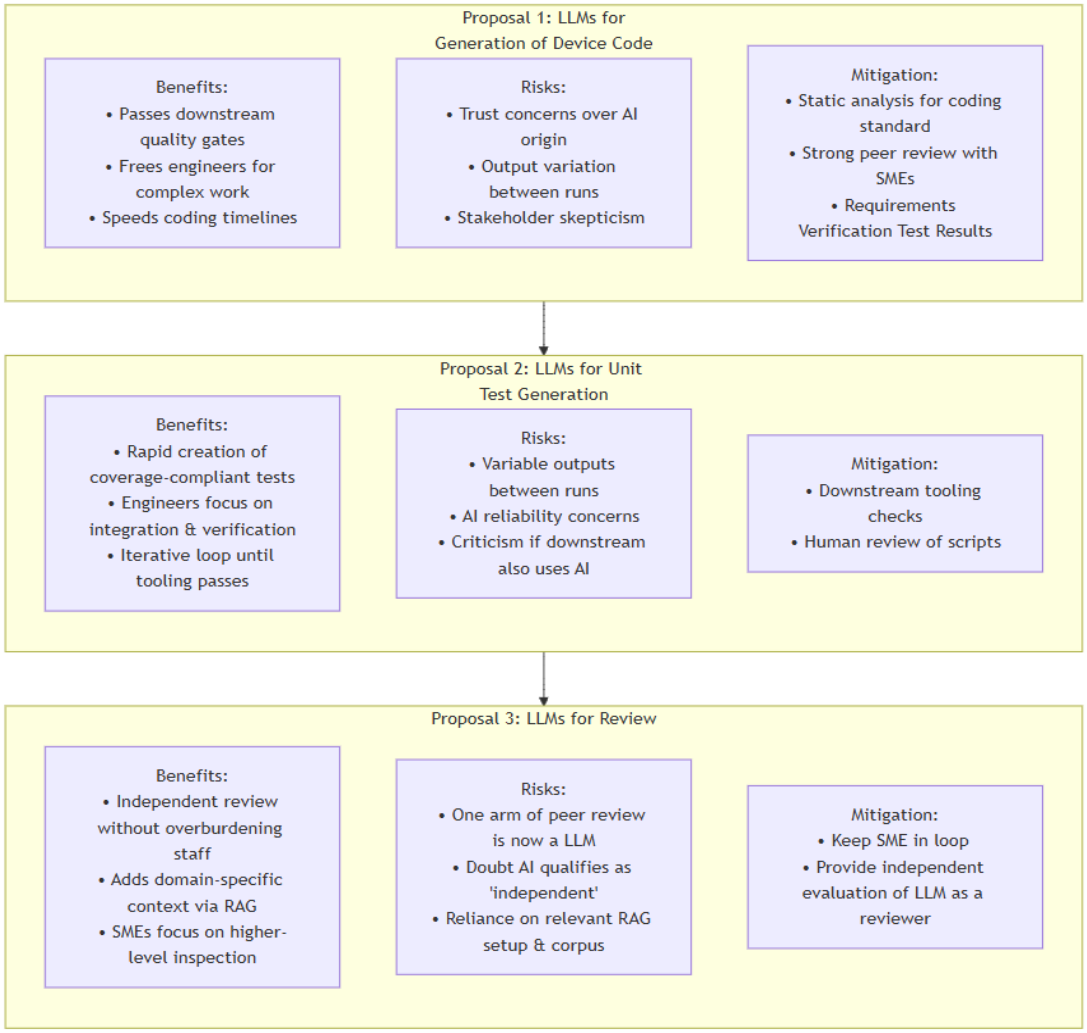

Figure 8

Conclusion

There is little doubt that AI will challenge the FDA's balancing act of protecting public health without impeding innovation in ways that are not easily anticipated. As citizens and patients in the health-care system, we should remain engaged and add our voices to the creation of democratic policymaking as the opportunities allow. As engineers involved in medical device development, hopefully the brief analysis and key proposals presented in this paper prove helpful to the discussion. The scope was limited to three proposals which will be the focus of upcoming research in 2026. The goal is to publish the MDM research findings for these proposals to further the conversation on the use of AI in medical device software development.

References

[1] *Report to Congressional Requesters, GAO-24-106122.* (2024, January). gao.gov. https://www.gao.gov/assets/d24106122.pdf

[2] *Federal Regulation: Selected Emerging Technologies Highlight the Need for Legislative Analysis and Enhanced Coordination.* (2024, January). gao.gov. https://www.gao.gov/products/gao-24-106122

[3] *Cybersecurity in Medical Devices: Quality System Proposals and Content of Premarket Submissions* (2025, June). fda.gov. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/cybersecurity-medical-devices-quality-system-proposals-and-content-premarket-submissions

[4] *Good Machine Learning Practice for Medical Device Development: Guiding Principles* (2021). fda.gov. https://www.fda.gov/medical-devices/software-medical-device-samd/good-machine-learning-practice-medical-device-development-guiding-principles

[5] *FDA Updates List of Artificial Intelligence and Machine Learning-Enabled Devices* (2023, October). fda.org. https://content.govdelivery.com/accounts/USFDA/bulletins/37671b1

[6] *Transparency for Machine Learning – Enabled Medical Devices: Guiding Principles* (2024, June). fda.org. https://www.fda.gov/medical-devices/software-medical-device-samd/transparency-machine-learning-enabled-medical-devices-guiding-principles

[7] *Marketing Submission Recommendations for a Predetermined Change Control Plan for Artificial Intelligence-Enabled Device Software Functions.* (2025, August). fda.org. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/marketing-submission-recommendations-predetermined-change-control-plan-artificial-intelligence

[8] *Public Law 117–328 117th Congress* (2022, December). congress.gov. https://www.congress.gov/117/plaws/publ328/PLAW-117publ328.pdf

[9] *White House Unveils America's AI Action Plan* (2025, July). whitehouse.gov. https://www.whitehouse.gov/articles/2025/07/white-house-unveils-americas-ai-action-plan/

[10] *Good machine learning practice for medical device development: Guiding principles*. (2025, January). imdrf.org. https://www.imdrf.org/sites/default/files/2025-02/IMDRF_AIML%20WG_GMLP_N88%20Final.pdf

[11] *Artificial Intelligence-Enabled Device Software Functions: Lifecycle Management and Market Submission Recommendations*. Draft Guidance. (2025, January). fda.org. https://www.fda.gov/media/184856/download

[12] *General Principles of Software Validation* (2002, January). fda.org. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/general-principles-software-validation

[13] *Computer Software Assurance for Production and Quality Management System Software.* (2026, February). fda.org. https://www.fda.gov/media/188844/download

[14] *Medical Devices; Quality System Regulation Amendments*. (2022, February). federalregister.gov. https://www.federalregister.gov/documents/2022/02/23/2022-03227/medicaldevices-quality-system-regulation-amendments

[15] *Medical Devices-Quality management systems-Requirements for Regulatory Purposes* (2016). ISO 134585:2016/(R) 2019. iso.org. https://www.iso.org/standard/59752.html