Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Regulatory-Grade AI Change Management: Versioning, Traceability, and Rollback for Iterative Algorithms in Medical Devices

Shrikant Chikhalkar

Independent Researcher, USA

ARTICLE INFO

Received: 04 March 2026

Accepted: 08 March 2026

ABSTRACT

Managing the change management process around AI-enabled software functions embedded within other medical device products requires more than deterministic software governance frameworks. Changing how businesses manage AI in medical devices needs more than just strict rules. Regular updates to an AI model—like changes to the training data, the way the model is built, its settings, or how it processes results—can create different and unexpected risks that might not be clear from overall performance numbers. So, the rules for managing changes should involve keeping a detailed record of all changes made, creating processes that can be repeated and checked with secure signed documents, and using clear step-by-step checks with set plans for what changes are allowed. Additionally, continuous monitoring at each deployment stage, ongoing model health surveillance, and clearly defined quantitative conditions for reverting changes are essential. Organizations must also establish structured rollback procedures with traceable documentation and require formal performance testing and approval of the model both before and after any changes or rollbacks are executed. These controls should be completely included in the standard operating procedures of the quality management system to make sure that AI-enabled software works clearly, can be checked, and stays in line with the necessary rules during the entire product lifecycle.

Keywords: AI-Enabled Devices, Change Control, Model Governance, Traceability, Staged Rollout

1. Introduction

AI-based software features have therefore become standard components of medical devices with clinical applications ranging from image enhancement for diagnostics to patient risk stratification. As a result, the global market for AI in health care is predicted to be worth USD 6.6 billion by 2021 at a compound annual growth rate (CAGR) of 40%, reflecting the pace at which machine learning is being embedded in regulated clinical products [1]. More advanced frameworks are now being developed by regulators such as the FDA, including the January 2021 AI/ML SaMD Action Plan, the October 2021 Good Machine Learning Practice principles, and the January 2025 Draft Guidance on AI-Enabled Device Software Functions. Fields such as imaging (e.g., radiology) and cardiovascular applications are the primary focus of these AI/ML-enabled medical devices [2].

Unlike deterministic software, the performance of AI may depend on the training data, model architecture, and testing conditions. An example is different performance when moving a model trained at one clinical site to other center data. Distribution shift may degrade performance in certain patient groups without affecting software functionality or logic. An uncontrolled model update may introduce vulnerabilities or diminish clinician trust. There are no code changes that customary software testing can measure to address these risks. According to estimates, in 2026, AI systems can help the U.S. healthcare system save USD 150 billion by reducing reactive treatment, focusing on

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

proactive health management, and preventing unnecessary hospitalizations, physician visits, and procedures from occurring in the first place [1].

Doing safe iteration with appropriate audit trails, patient safety, and regulatory defensibility means that the combination of change control plans that are defined by the FDA in its December 2024 Final Guidance and its April 2023 Draft Guidance on Marketing Submission Recommendations will provide prior notice of what changes to expect, what validation will be used, and what post-market surveillance will be used to determine whether the system is still being deployed [2]. This requires a change management framework to specify what can change, what evidence is created as changes are made, and how releases can be rolled back if monitoring after deployment reveals increased risk.

2. Characterizing Change Types and Risk Profiles

In general, different parts of the AI-enabled decision support system can change together, such as how training data is organized and labeled, how data is prepared, the features and models used, how decisions are adjusted, and how results are evaluated. The risks associated with these different categories are neither homogeneous nor predictable from the overall category. In a well-known study of the COMPAS recidivism algorithm, which was tested on 10,000 criminal defendants in Broward County, Florida, the overall accuracy of the system was 61% for general recidivism and 20% for violent recidivism, showing that a predictive system can perform well overall even if it does a poor job predicting one specific type.

Other possible influences include the training data distribution and the nature of the cohorts. In a cohort without stratification, a scoring model may perform differently based on the demographic strata. Although concordance rates were similar for White (62.5%) and Black (62.3%) individuals, the pattern of error in COMPAS [4] was asymmetric. Black defendants predicted to recidivate when they did not reoffend were rated as higher risk, almost twice as much as white defendants who did not recidivate. White defendants who reoffended were classified as low risk at twice the rate as Black reoffenders (48% to 28%) [3].

A defensible change management process should define change types and pre-specify requirements for evidence types and amounts to ensure that validation is commensurate with change size and type. Systems tested mainly on unrepresentative groups may have biased errors that standard accuracy measures can't find and are unlikely to work well for underrepresented groups. Further, the COMPAS dataset illustrates that stratification is necessary to detect subgroup-level performance divergence. The overall concordance rate is 63.6%, but this masks the meaningful variation in false positive rates and false negative rates across the demographic subgroups [4]. If there is a small change in the threshold, testing may only focus on specific subgroups; however, bigger changes to the model will need a full review of how it performs based on set acceptance standards for different demographic groups.

Context	Metric	Subgroup / Condition	Value
COMPAS Study Sample	Total defendants tested	Broward County, Florida	10,000
COMPAS Predictive Accuracy	Overall accuracy – general recidivism	All defendants	61%
	Overall accuracy –	All defendants	20%

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

	violent recidivism		
COMPAS Concordance Rate	Concordance rate	White individuals	62.50%
	Concordance rate	Black individuals	62.30%
	Overall concordance rate	All defendants	63.60%
COMPAS Error Pattern	White defendants classified as low risk who reoffended	White reoffenders	48%
	Black defendants classified as low risk who reoffended	Black reoffenders	28%

Table 1: Subgroup Performance Disparities and Accuracy Metrics in COMPAS Recidivism Algorithm Risk Profiling [3, 4]

3. Lineage Tracking and Reproducible Pipelines

The first step of regulatory-grade change management is a commitment to reproducibility at the infrastructure level: every release of a model must specify the code version, pinned software dependencies, training configuration, and dataset snapshot it was trained on. This is not merely a procedural burden. Sculley et al. show that for mature ML systems, only 5% of system code is machine learning, and the other 95% is data pipelines, glue code, and configuration infrastructure [5]. So the above property means that reproducibility problems will often arise in the pipeline, rather than the model code, which has implications for regulated software. For example, reconstructing the state of a model is important in post-market surveillance or when investigating adverse events.

Based on this, tracking dataset lineage using dataset registries that can track dataset fingerprints, reason about provenance labels, and use governance-based audit approval for data inclusion is needed to ensure future data audits can reconstruct what was used and under what curation policies, as the technical debt principle CACE (Changing Anything Changes Everything) suggests that seemingly innocuous undocumented changes to input signals or preprocessing configurations can reverberate throughout a model's behavior [5].

Before being uploaded to a secure model registry, model artifacts like the model binaries, preprocessing graph, and calibration parameters are signed with a cryptographic method to ensure they can be checked later if something goes wrong and we need to find out what happened. This control is needed, as supply chain security research found that the median time in which embedded malicious packages went undetected before public disclosure was 209 days, and the maximum time was 1216 days across the analyzed incidents [6]. 61% of the analyzed malicious packages used tampering through typosquatting. and 49% used obfuscation to evade detection in the same time range [6]. Build systems should record tamper-clear audit logs that detail every transformation event performed on a source artifact as it is transformed into a deployed model. Role-based access controls should be implemented to provide separation of duties to model development, validation, and deployment authorization teams. Together, these controls mean that the answer to the question "Why was this model released?" is auditable through the pipelines.

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

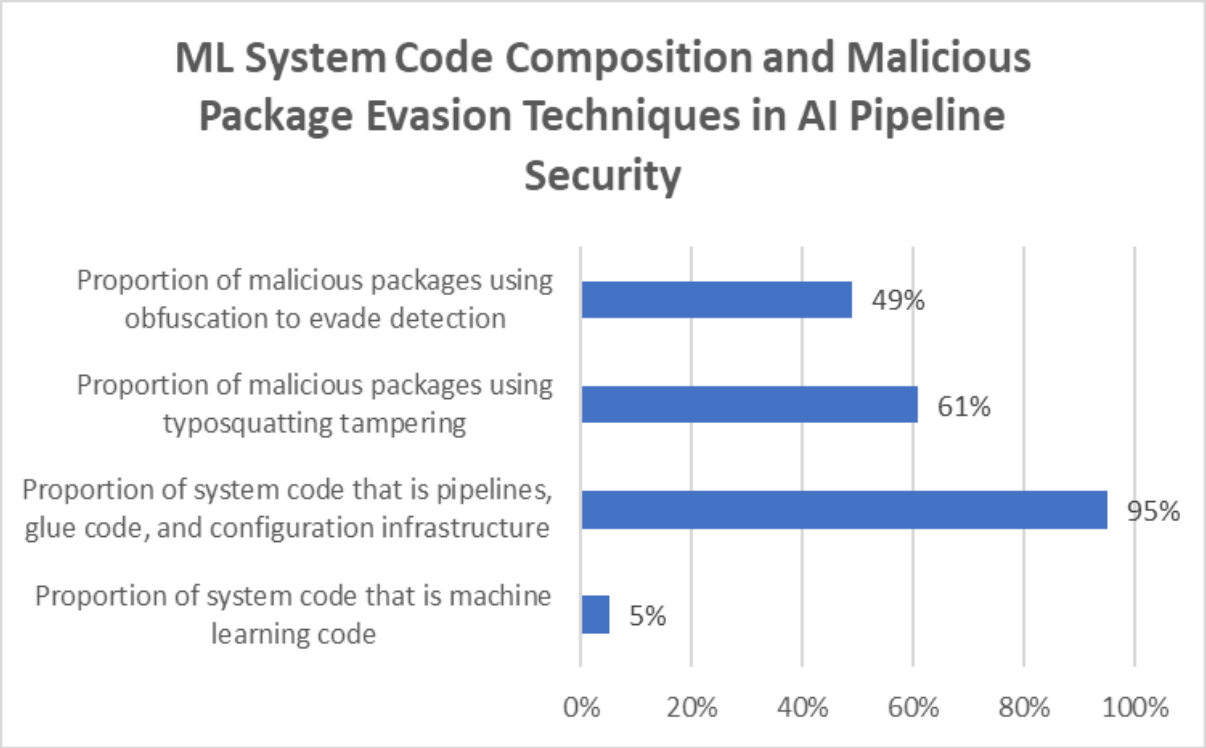

Fig 1: Proportional Distribution of ML Infrastructure Code and Supply Chain Attack Vectors in Reproducible Pipeline Governance [5, 6]

4. Verification, Validation, and Predetermined Change Planning

Each model release should be evaluated on held-out test data. Cross-site validation should also be considered, as should stratified performance in clinically relevant subgroups. The work of Dressel and Farid is illustrative of the importance of subgroup stratified evaluation in algorithm assessment. Despite COMPAS as a whole being 65% accurate, both the false positive rate and false negative rates were unsymmetrically dispersed across racial subgroups, with a false positive rate of 40.4% for Black defendants compared to 25.4% for white defendants and a false negative rate of 30.9% for Black defendants compared to 47.9% for white defendants [7]. Simply removing race from the input features did not fix these imbalances, showing that overall accuracy can hide problems in how well the algorithm works for different groups, which a detailed evaluation would uncover.

The same study reported that the 137 features that COMPAS used performed no better than a simple classifier of the age of the person and the total number of prior convictions, which achieved 65-66% accuracy as well. These findings suggest certain implications for validation: validation should examine whether complex features contribute additional performance; otherwise, simple models should be preferred to larger models to reduce the risk of overfitting to the specifics of the datasets.

Robustness testing requires realistic input perturbation (noise, protocol, and apparatus variances) and stress testing variants from risk analysis tools identifying and synthesizing potential failure modes. A different prediction method using input from just 20 participants in each group reached an accuracy of 67.0% and a 0.71 AUC-ROC level, which is very similar to COMPAS's score. These analyses show that predictive ceiling effects constrain model performance regardless of algorithmic sophistication [7].

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The change management plan outlines the process for implementing safe iterations, including the types of changes that can be expected, how they will be validated, and what post-market surveillance will be performed. The FDA has three levels of risk for AI/ML-enabled software as a medical device: Class I (low risk), Class II (moderate risk), and Class III (high risk). Each level has more rules and checks. [8] A predetermined change control plan pathway enables manufacturers to prospectively define the limits of changes. This method will make ongoing improvements a regular process with clear and checkable rules, while also making it easier to submit changes that meet set validation standards. [8]

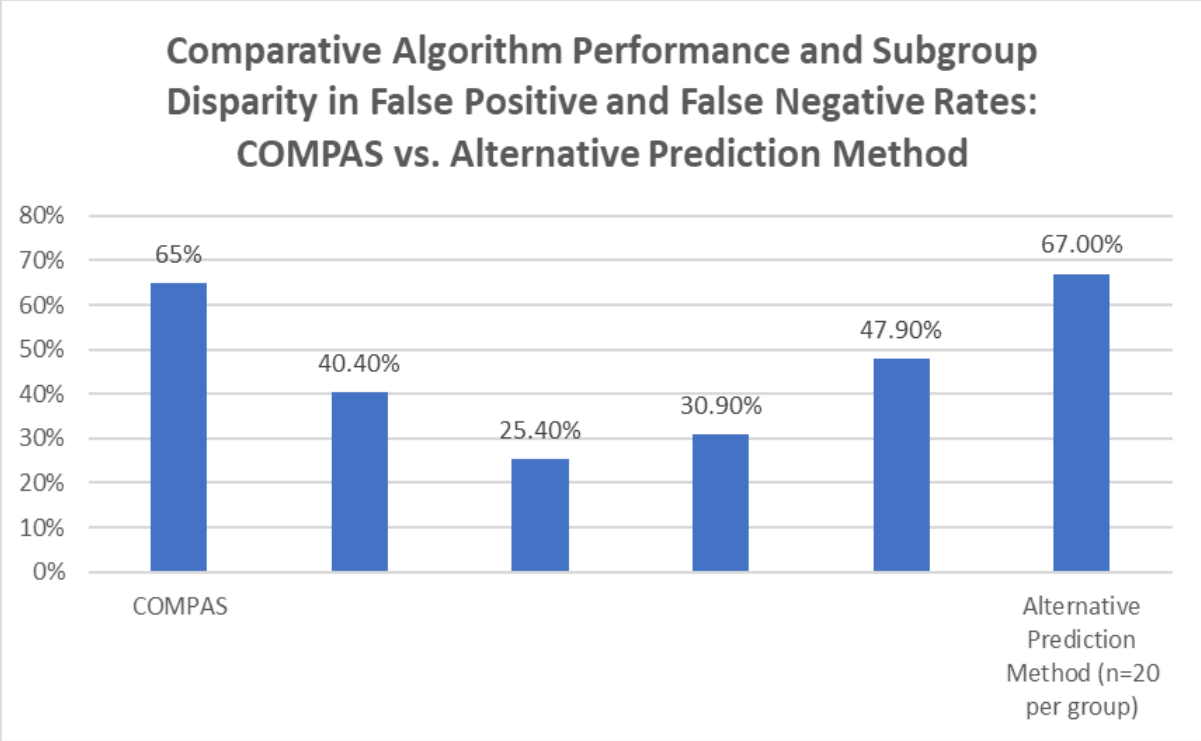

Figure 2 shows the differences in error rates for different groups and the overall accuracy of COMPAS and other classifier models [7, 8]

5. Staged Deployment, Monitoring, and Rollback

A phased rollout consists of successive stages between an internal validation environment, a limited-release pilot, and a larger gradual rollout, with monitoring gates between each stage. The same approach of gradual rollouts, with monitoring checks, is commonly used in high-assurance software engineering continuous delivery to lower the chances of problems during production by 40%–60% compared to launching everything at once, because each phase limits the impact of any hidden issues to a smaller group of users. [9] Feature controls also make it easier to turn off or go back on AI outputs without needing to reinstall the software, which speeds up fixing issues and avoids problems with regulations that come from having to release the entire software again.

After the software is in use, we keep an eye on various technical signals (like delays, error rates, and system uptime) and model health signals (such as changes in input data, output patterns, and unusual alert rates). Tools for detecting dataset shifts, such as the population stability index and statistical divergence metrics (like the Kullback-Leibler divergence), identify changes in the data coming into the

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

inference pipeline. These tools can detect changes in the main features of the data when 10% or 15% of a typical group's population shifts, even before any noticeable drop in performance occurs. Pipelines must also keep structured, timestamped records of each event that triggers a flag for audit or regulatory review.

Setting clear rules for when to reverse changes, based on specific measurements like a big drop in performance or a long-lasting rise in alerts, will automatically start the process to fix issues, making the rollback process more efficient. In Production AI Deployment programs, organizations with clear rollback criteria common across all teams have seen mean rollback times under four hours, compared to over 24 hours in unstructured rollback processes [10]. This rollback needs to be quick, completely trackable, and checked against a reliable system backup, not just the last known state, with tests after the rollback to ensure that performance is back to the expected level.

Deployment Stage	Monitoring Category	Key Metric / Signal	Threshold/Benchmark	Outcome / Benefit
Phased Rollout	Deployment Architecture	Production Incident Rate Reduction	40%–60% reduction vs. all-at-once deployment	Limits blast radius of undetected defects
	Feature Controls	Mean Time to Remediation	Reduced (no full software reinstall required)	Eliminates regulatory concerns from full build reissuance
Post-Deployment	Technical Health	Latency, Error Rates, Uptime	Baseline thresholds per system SLA	Early detection of infrastructure degradation
	Model Health	Input Data Drift	Population Stability Index (PSI)	Detects distributional shift before performance drop
	Model Health	Output Distributional Drift	Kullback-Leibler (KL) Divergence	Detects shift when 10%–15% of cohort population changes
	Model Health	Alert Rate Anomalies	Sustained increase above normal baseline	Triggers escalation and remediation processes
	Audit & Compliance	Timestamped Event Logs	Structured records per flagged event	Supports audit and regulatory review traceability
Rollback	Rollback Efficiency	Mean Rollback Time (Structured)	< 4 hours	Achieved with clear, cross-team

				rollback criteria
	Rollback Efficiency	Mean Rollback Time (Unstructured)	> 24 hours	Baseline for organizations without defined rollback criteria
	Rollback Validation	Post-Rollback Performance Check	Performance restored to expected baseline level	Verified against reliable system backup, not last known state

Table 2: Staged Deployment Monitoring Metrics, Rollback Criteria, and Performance Benchmarks for AI-Enabled Postmarket Surveillance Platforms [9, 10]

Conclusion

Regulatory AI change management should be an ongoing, careful, and data-focused process where the software for AI devices is expected, described, and tested against clear performance standards before it is released, instead of waiting to see if there are problems after release that could negatively affect patient care. The combination of tracking data from start to finish, ensuring results can be repeated, validating in different groups, gradually introducing changes, monitoring how well the model works, and having a way to revert changes By building on existing quality systems, the architecture provides a platform for external audit and instills confidence amongst practitioners that this form of system may be deployed in regulated clinical practice. In the future, the speed and safety of deploying validated improvements will depend more on the governance of change than on the performance of the underlying models, as the capability and AI-based functionality in medical devices become more embedded.

References

[1] Adam Bohr and Kaveh Memarzadeh, "The rise of artificial intelligence in healthcare applications," National Library of Medicine, 2020. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7325854/
[2] U.S. Food and Drug Administration, "Artificial Intelligence in Software as a Medical Device," 2025. [Online]. Available: https://www.fda.gov/medical-devices/software-medical-device-samd/artificial-intelligence-software-medical-device
[3] Jeff Larson et al., "How We Analyzed the COMPAS Recidivism Algorithm," ProPublica, May 2016. [Online]. Available: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm
[4] Samuel G. Finlayson et al., "Adversarial attacks on medical machine learning: Emerging vulnerabilities demand new conversations," HHS Public Access, 2019. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7657648/pdf/nihms-1642476.pdf
[5] D. Sculley et al., "Hidden Technical Debt in Machine Learning Systems." [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf
[6] Marc Ohm et al., "Backstabber's Knife Collection: A Review of Open Source Software Supply Chain Attacks," Springer, 2020. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7338168/pdf/978-3-030-52683-2_Chapter_2.pdf
[7] Julia Dressel and Hany Farid, "The accuracy, fairness, and limits of predicting recidivism," 2018. [Online]. Available: https://www.science.org/doi/pdf/10.1126/sciadv.aao5580

[8] Liron Pantanowitz et al., "Regulatory Aspects of Artificial Intelligence and Machine Learning," Modern Pathology, Volume 37, Issue 12, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893395224001893
[9] Eric Breck et al., "The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction," 2017. [Online]. Available: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/aad9f93b86b7addfea4c419b9100c6cdd26cacea.pdf
[10]Satvik Garg et al., "On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps," arXiv:2202.03541v1, 2022. [Online]. Available: https://arxiv.org/pdf/2202.03541