International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Artificial Intelligence-Driven Software Validation Engineering Framework for Improving Digital Healthcare Applications Supporting Individuals with Autism Spectrum Disorder

Itendra Kumar Singh[1]

Abstract: Digital health tools built for individuals with Autism Spectrum Disorder (ASD) increasingly rely on machine learning components that continue to adapt after deployment, behavioral-monitoring sensors, augmentative communication platforms, and clinician-facing decision-support dashboards, among them. Conventional software validation practice, built around a fixed pre-release test surface, was not designed to govern systems whose decision logic shifts as new data arrive. This paper proposes a validation-engineering framework tailored to AI-enabled ASD digital-health applications, treating continuous verification, explainability, accessibility, and cybersecurity as engineering disciplines rather than downstream compliance items. The framework layers a modular, human-in-the-loop validation architecture over existing medical-device software standards, translating IEC 62304 and FDA AI/ML lifecycle guidance into an operational validation pipeline suited to systems that keep learning. Drawing on the software-validation-engineering and AI-in-ASD-care literature, two bodies of work that have so far developed largely apart from one another, the paper argues that a distinct engineering discipline, not clinical or regulatory oversight alone, should anchor trust in these systems. The architectural, infrastructural, and performance layers of the proposed framework are evaluated against accessibility and equity considerations specific to neurodiverse users, and the paper closes by identifying validation-engineering research questions this framework does not yet resolve.

Keywords: *Software Validation Engineering, Artificial Intelligence, Autism Spectrum Disorder, Digital Health Care, Explainable AI, Accessibility, Cybersecurity, Human-in-the-Loop Governance*

1. Introduction

Autism Spectrum Disorder affects communication, social interaction, and sensory processing in ways that vary substantially from one individual to the next, and clinical services have struggled for decades to scale personalized support to the population that needs it. Digital health platforms, behavioral-therapy applications, augmentative and alternative communication (AAC) tools, computer-vision-based monitoring systems, and telehealth portals have expanded access to intervention in settings where specialist care is scarce or costly. Machine learning components embedded in these platforms now personalize therapy pacing, flag behavioral changes from wearable sensor streams, and support caregivers with predictive alerts (Deng & Rattadilok, 2022; Ahuja et al., 2022).

The engineering discipline responsible for confirming that software behaves as intended was built for a different kind of system. Test suites, code reviews, and release gates in conventional software engineering assume that a system's decision logic is fixed at deployment and changes only through a traceable, versioned update. Machine learning models embedded in ASD-focused applications frequently retrain on new behavioral or sensor data, and their outputs can shift in ways that are neither documented in a changelog nor visible to the validation team that signed off on the previous release. A model that classified a sensory-overload episode reliably in one deployment window may drift by the next without any code change triggering review.

Software validation engineering has traditionally addressed this gap through static test coverage, regression suites, and pre-release sign-off approaches that presuppose a validation target holding still long enough to be tested. Standards bodies have started to respond: the U.S. Food and Drug Administration's AI/ML-Based Software as a Medical Device Action Plan (2021) calls for a predetermined change-control plan for adaptive algorithms, and IEC 62304 continues to anchor medical-device software lifecycle obligations even though its drafters did not anticipate continuously retraining models. Neither instrument specifies how a validation team should operationalize continuous verification, accessibility testing, or explainability auditing for a system whose behavior is itself in motion.

Enterprise software validation practice outside healthcare already treats continuous integration and continuous validation (CI/CV) as inseparable pipelines that gate a release on automated test batteries, security scans, and performance regressions before a human reviewer ever sees the change. Healthcare AI validation has been slower to adopt this posture, partly because clinical risk tolerance is lower and partly because the accessibility and explainability requirements specific to a neurodiverse user base do not map cleanly onto conventional CI/CV test batteries built for generic web or enterprise software. Extending that discipline to ASD-focused AI systems requires deciding what a continuous validation gate should actually check, who reviews a failed gate,

Independent Researcher,USA

itendrasingh108@gmail.com

and how findings feed back into both the software release process and the standards documentation regulators expect.

The literature on AI applications for ASD care has grown quickly along a different axis. Reviews of AI-enabled ASD interventions catalogue clinical outcomes, diagnostic accuracy, and caregiver acceptance (Sánchez de las Heras et al., 2026; Rajpurkar et al., 2022); however, they evaluate these systems as clinical tools rather than as software artifacts requiring an engineering validation discipline of their own. Explainability research in clinical decision support (Van Dort et al., 2026; Liu et al., 2024) and machine-learning validation research in adjacent regulated domains such as the clinical laboratory (Spies et al., 2024; Boman, 2023) point toward the components such a discipline would need, but no existing framework assembles them specifically for accessibility-sensitive, continuously-learning ASD applications.

This paper proposes a validation-engineering framework built to close that gap. The framework treats architecture, infrastructure, and performance as three interlocking validation layers rather than as separate development concerns bolted onto a testing checklist at the end of a release cycle. Section 2 reviews the AI-in-ASD-care, clinical-AI-governance, and software-validation-standards literatures that motivate this synthesis. Section 3 develops the framework across its architectural, infrastructural, and performance dimensions. Section 4 considers the governance, equity, and privacy implications of operationalizing continuous validation for a neurodiverse user population, and Section 5 closes with the framework's limitations and the validation-engineering research questions it leaves open.

2. Related Work

2.1. AI Applications in ASD Care

Sensory-state classification from wearable sensor data, behavioral-pattern detection from video or motion capture, and adaptive scheduling of therapeutic content are just a few of the several problem categories in which machine learning has been used to help ASD.

Sensory-state classification from wearable sensor data, behavioral-pattern detection from video or motion capture, and adaptive scheduling of therapeutic content are just a few of the several problem categories in which machine learning has been used to help ASD.

Deng and Rattadilok (2022) created a sensor-and-machine-learning recommendation system that uses physiological signals to classify attention and stress states in children with ASD and then modifies a sensory-management intervention accordingly. The system was validated with a comparison group of typically developing children as well as an ASD cohort, and it included a retraining cycle meant to increase classification accuracy as more data accumulated. The majority of published systems report classification performance without a systematic account of how that performance holds up after deployment, according to Ahuja et al.'s (2022) review of wearable technology for tracking behavioral and physiological responses in ASD populations. They also cataloged the sensor modalities in use. Sánchez de las Heras et al. (2026) synthesized AI approaches that serve individuals with ASD throughout the areas of evaluation, intervention, and communication. They came to the conclusion that rather than coming together on common engineering practice, the field is still divided across disparate pilot systems.

Accuracy on a held-out test set and a usability score from a pilot cohort are the artifacts this literature treats as validation. Rajpurkar et al. (2022) noted the same pattern across health AI more broadly: clinical studies establish that a model performs adequately at a point in time, but the engineering practices needed to keep that performance stable across deployment environments and data drift receive comparatively little attention once the clinical trial ends.

The methodological pattern across these systems is worth examining more closely. Deng and Rattadilok (2022) validated their sensory-management system against both an ASD cohort and a typically developing comparison group specifically to establish that classification accuracy did not simply reflect population-level differences in physiological signal patterns, a design choice that strengthens the clinical claim but still stops at the point of initial deployment rather than tracking the system across a longer operational period. Ahuja et al. (2022) found the same boundary across the wearable-technology literature they reviewed: sensor modalities such as electrodermal activity, heart-rate variability, and accelerometry are well characterized individually, but the review found comparatively few studies reporting how sensor-fusion models performed once multiple modalities were combined into a single classifier deployed outside a controlled study setting.

Sánchez de las Heras et al. (2026) organized the broader AI-for-ASD literature into assessment, intervention, and communication-support categories, and their fragmentation finding is more specific than a general observation about disconnected pilot studies: validation methodology varied enough between categories—diagnostic-support tools reporting sensitivity and specificity, intervention tools reporting behavioral outcome measures, and communication tools reporting usability scores—that no shared vocabulary yet exists for describing what "validated" means across the field. Rajpurkar et al. (2022) reached a parallel conclusion for health AI generally, arguing that the absence of standardized post-deployment monitoring practice, rather than a shortage of promising models, is what currently limits how much clinical AI research translates into dependable production systems.

2.2. AI Applications in ASD Care

Parallel literature has examined how clinicians and health systems build trust in AI-generated recommendations. Van Dort et al. (2026) conducted a scoping review of healthcare professionals' perspectives on explainable AI in hospital clinical decision support. Sixteen studies met their inclusion criteria, finding that clinicians want explanations that map onto their existing diagnostic reasoning rather than technical feature-importance outputs and that explanation quality shapes willingness to act on a system's recommendation more than raw accuracy does. Liu et al. (2024) demonstrated an approach for using explainable AI methods to identify which features drive a clinical decision-support model's alerts, then used that information to prune low-value alerts and reduce clinician alert fatigue, a validation activity that sits between model development and deployment monitoring and that few existing frameworks assign clearly to either phase.

In support of human-curated validation of machine learning algorithms applied to health data, Boman (2023) suggested that domain-expert assessment of model outputs be incorporated into the validation pipeline as opposed to being handled as an ad hoc

override mechanism.

This reasoning was extended to the clinical laboratory by Spies et al. (2024), who described a machine learning solution validation, implementation, and monitoring lifecycle that includes analytical validation, continuous performance monitoring, and specified criteria for when a deployed model needs to be re-validated. Amann et al. (2020) took a multidisciplinary approach to explainability in healthcare AI, contending that explainability has different functions for regulators, patients, and clinicians and that a validation framework that views explainability as a single, undifferentiated requirement will not adequately satisfy any of these audiences.

Van Dort et al. (2026) documented a particular practical consequence for how a validation team should design its explainability audits: an audit that merely verifies that a model generates a technically correct feature-importance output, without examining whether that output maps onto the diagnostic categories a clinician actually reasons in, would pass a narrow correctness test while failing the adoption outcome that the explainability requirement exists to support. A version of this issue was operationalized by Liu et al. (2024), who treated alert fatigue as a validation signal rather than a downstream usability complaint handled independently from model performance. They did this by using explainability output to identify and prune the specific alert types that clinicians were overriding most frequently.

Spies et al. (2024) and Boman (2023) converge on a similar structural point from different regulated domains: validation is not a gate a model passes once but a lifecycle with defined triggers for re-validation. Spies et al. (2024) specify criteria drawn from clinical laboratory practice, a shift in patient population, a change in specimen-collection equipment, and a scheduled interval that would each independently trigger a revalidation cycle for a deployed model. Amann et al. (2020) add that these triggers cannot be defined by engineering criteria alone, because what counts as a meaningful population shift for a clinician differs from what counts as a meaningful shift for a regulator auditing compliance documentation, and a validation framework that picks one audience's definition by default will underserve the other.

This literature establishes the governance components a validation framework for AI-enabled ASD applications needs: human review, explainability, and monitoring. None of it, however, addresses the accessibility dimension specific to a neurodiverse user population, and none connects these governance components to a concrete software architecture.

2.3. Software Validation Standards and Medical-Device Regulation

The software engineering literature offers the structural half of the picture. The IEEE Software Engineering Body of Knowledge (SWEBOK v4.0) and standard references such as Pressman and Maxim (2020) and Sommerville (2021) define validation as confirming that a system meets its intended use, distinct from verification's narrower question of whether the system was built to specification. ISO/IEC/IEEE 29119 formalizes software testing processes and documentation, and IEC 62304 governs the medical-device software lifecycle, requiring risk-based classification, traceability from requirements to test evidence, and change-control documentation. ISO 13485 layers a quality management system obligation on top of these lifecycle requirements for organizations that manufacture medical device software.

The gap identified above is not merely definitional. IEC 62304's risk-based classification scheme assigns a software safety class at the design stage and expects traceability artifacts to remain valid through the product's operational life; a model that shifts classification behavior through retraining effectively changes its own risk profile after that classification was assigned, without a defined mechanism in the standard for revisiting the assignment. Pressman and Maxim (2020) and Sommerville (2021) both describe validation as confirming intended use against a requirements baseline, and a continuously retraining model has no single requirements baseline to validate against; it has a moving target that the baseline itself would need to be re-derived for at each retraining event, a re-derivation neither text's validation lifecycle anticipates.

A validation target that does not change between audits is what these compliance artifacts assume: traceability matrices, static test plans, and fixed acceptance criteria. Continuously retraining models violate that assumption by design. The FDA's AI/ML-Based Software as a Medical Device Action Plan (2021) acknowledges the gap directly, proposing a predetermined change control plan that would let a manufacturer pre-authorize certain categories of model update without full resubmission, though the Action Plan stops short of specifying the engineering practices, test automation, monitoring infrastructure, and rollback procedures that would make such a plan operationally credible rather than aspirational. Table 1 summarizes how far each instrument reaches toward the three capabilities a continuously learning ASD application actually needs.

Standard/ Guidance	Primary Scope	Addresses Continuous Model Change	Addresses Neurodiverse Accessibility	Addresses Cybersecurity Monitoring
ISO/IEC/IEEE 29119	Software testing process & documentation	No	No	No
IEC 62304	Medical-device software lifecycle	Partial (change control, not adaptive models)	No	Partial (risk management only)
ISO 13485	Quality management system	No	No	No
FDA AI/ML Action Plan (2021)	AI/ML-based SaMD lifecycle	Partial (predetermined change control plan)	No	No

Cybersecurity adds a further dimension that the validation-engineering literature has begun to quantify empirically. 106 of the 201 software and medical device vulnerabilities that were revealed between 2001 and 2022 were categorized as critical for health by Mejía-Granda et al. (2024). In a national-scale analysis

of purchased medical devices, Bracciale et al. (2023) discovered a similar severity pattern, finding 661 unique vulnerabilities with an average exposure window of about 3.2 years between device deployment and public vulnerability disclosure, more than half of which were rated critical or high-severity. While neither study focuses on ASD applications specifically, they both deal with the same class of sensor-integrated, continuously connected health software that this paper discusses.

Foundational artificial intelligence references (Russell & Norvig, 2021; Goodfellow et al., 2016) and Topol's (2019) account of AI's convergence with clinical medicine establish the technical vocabulary model drift, retraining, and representation learning this framework relies on, without themselves addressing software validation practice. The gap this paper responds to sits precisely at the intersection these three literatures leave open: engineering standards without an adaptive system operating model, clinical AI governance without a software architecture, and ASD-specific AI research without either.

3. Proposed Framework

3.1. Architectural Paradigms and Sustainability

Four architectural layers, not a single test-and-release gate, organize validation responsibility in the proposed framework. A modular validation services layer decomposes validation into independently deployable services: model-performance validation, accessibility validation, explainability auditing, and security scanning, mirroring the way enterprise systems decompose business capability into microservices. Each validation service owns its own test corpus and can be updated, retrained, or replaced without redeploying the others, which matters in practice because an accessibility-testing update (a new WCAG guideline, for instance) should not require re-running an unrelated security scan.

Each validation service inside this layer is domain-specific rather than generic. An accessibility validation service built for an augmentative communication tool tests visual-symbol legibility, response-time consistency, and predictable button placement, while the equivalent service for a wearable sensor application tests alert timing, vibration-pattern consistency, and battery-drain thresholds that could otherwise cause a device to fail silently during a monitoring window. Collapsing these into one generic accessibility test suite would satisfy a compliance checklist while missing the failure modes each application type actually presents to a neurodiverse user.

Layer ownership matters as much as layer boundaries. Assigning the Modular Validation Services layer to a dedicated validation-engineering function, rather than splitting accessibility, security, and performance testing across whichever development team happens to touch a given code path, is what keeps each service's test corpus coherent over time. Enterprise environments that rely on siloed, team-by-team testing where the team building a feature also decides how thoroughly to test it tend to accumulate exactly the kind of inconsistent validation coverage this layer is designed to prevent, because no team is incentivized to test its own accessibility or security debt as rigorously as an independent validation function would.

Above these services sits a Continuous Integration/Continuous Validation (CI/CV) layer that gates every model or code change against them before release. Where conventional CI/CD pipelines check that code compiles and passes unit tests, the CI/CV layer also checks that a retrained model's classification behavior has not drifted outside an approved tolerance band, that accessibility test batteries still pass against updated interface elements, and that no new dependency introduces a known vulnerability. A change that fails any of these gates is held for human review rather than deployed automatically, a practice already standard in regulated financial and telecommunications systems, and one that transfers naturally to healthcare software once the relevant test batteries exist.

The CI/CV layer's tolerance bands need to be set deliberately rather than inherited from generic software regression thresholds. A drift tolerance calibrated for a payment-processing system, where a one-percentage-point accuracy change might be immaterial, is not automatically appropriate for a sensory-overload classifier, where the same percentage change could translate into a specific number of missed alerts during exactly the moments an alert was designed to catch. Setting these bands requires the validation team to translate a clinical or accessibility consequence into a statistical threshold before the pipeline can enforce anything meaningfully. A translation step this framework treats as a deliverable of the Modular Validation Services layer rather than an assumption baked into the CI/CV layer's default configuration.

Below both layers, a Human-in-the-Loop (HITL) governance layer routes any validation-gate failure, along with a defined subset of low-confidence model outputs, to clinician or accessibility-specialist review before the system acts on them. This layer is deliberately positioned as an architectural component rather than an informal escalation path: it has its own service-level agreement for review turnaround, its own audit log, and its own criteria for when repeated low-confidence outputs on a particular feature should trigger a model-level investigation rather than case-by-case override. Boman's (2023) argument for human-curated validation and Van Dort et al.'s (2026) finding that clinicians want explanations tied to their existing reasoning both point toward this kind of structured escalation rather than an unstructured "human in the loop" label.

An interoperability layer completes the architecture, connecting validation to external systems, electronic health records, telehealth platforms, and caregiver-facing applications through standards such as HL7 FHIR, so that validation findings (a flagged sensor anomaly or a low-confidence therapy recommendation) can be surfaced to the clinical systems where a caregiver or clinician would actually see them, rather than remaining inside an internal validation dashboard no one outside the engineering team reviews. Figure 1 sets out how a single model or code change moves through these four layers.

Step	Layer	Action	Gate Outcome
1	Modular Validation Services	Change submitted; performance, accessibility, explainability, and security services each run their own test corpus	Pass → Step 2 / Fail → Step 3

2	CI/CV	Drift tolerance, accessibility regression, and dependency-vulnerability checks evaluated against approved bands	Pass → Step 4 / Fail → Step 3
3	Human-in-the-Loop Governance	Failed change or low-confidence output routed to clinician / accessibility-specialist review under a defined SLA	Approved → Step 4 / Rejected → returned to development
4	Interoperability	Validated change (or flagged finding) released and surfaced to EHR, telehealth, and caregiver-facing systems via HL7 FHIR	Deployment / notification complete

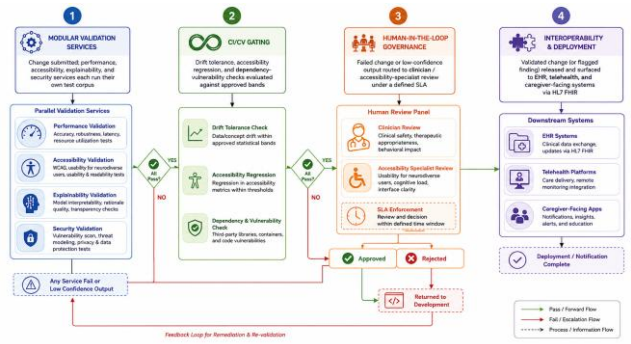

Fig. 1. Proposed continuous validation lifecycle: sequential flow of a model or code change through the framework's four architectural layers.

None of these four layers requires healthcare organizations to abandon existing enterprise tooling. Regulated financial and telecommunications environments already run comparable CI/CD pipelines with mandatory security gates and change-advisory review; the architecture proposed here layers model-specific drift checks, accessibility test batteries, and clinician escalation paths onto that same operational foundation rather than replacing it. Healthcare IT departments already maintain pipelines for conventional software, and asking them to adopt an entirely separate toolchain for AI components would slow adoption more than the underlying validation problem justifies.

Decomposition, not any single component, is what makes this architecture sustainable. Because validation services are independent, a regulatory change affecting only accessibility testing, a revised guideline for cognitive-load thresholds, say, can be absorbed by updating one service without re-certifying the entire pipeline. This containment of change is the same property that has made microservice architectures durable in other regulated enterprise contexts, translated here to a validation-specific purpose.

3.2. Infrastructure Optimization for Continuous Validation

A validation architecture that only exists on paper protects no one. Running the four layers above continuously rather than at fixed release milestones requires infrastructure built for that cadence. Container orchestration (Docker, Kubernetes) lets each validation service scale on its own terms. Accessibility testing, triggered mainly by interface changes, carries a very different load profile than security scanning, which should run against every dependency update and let a failing service roll back without dragging the others down with it.

Infrastructure choices in this layer are not independent of the architectural decisions in Section 3.1. A Modular Validation Services layer that treats accessibility and security as separate services only delivers its intended benefit if the underlying infrastructure actually deploys and scales them separately; running all four validation services on a shared, undifferentiated compute pool reintroduces the coupling the architecture was designed to eliminate, since a spike in security-scanning load would then compete for the same resources an accessibility-testing run needs, recreating the cross-service interference the layered architecture exists to prevent.

Observability is the layer most existing healthcare AI deployments underinvest in relative to its importance for this framework. Distributed tracing and structured logging across the validation services let an engineer reconstruct, after the fact, exactly which validation checks a given model version passed and which low-confidence outputs were routed to human review, a record that matters both for internal debugging and for the traceability documentation IEC 62304 and ISO 13485 require. A validation-engineering framework without this instrumentation is a paper description of intent rather than an auditable practice.

Retention policy is an infrastructure decision with direct bearing on the traceability documentation IEC 62304 and ISO 13485 expect. Observability data logs, traces, and validation-gate outcomes needs to persist long enough to support an audit that may occur well after a given model version has been superseded, which means infrastructure sizing decisions cannot be made purely on current operational needs. A validation team that optimizes storage costs by discarding older observability data too aggressively may find itself unable to reconstruct why a now-retired model version passed or failed a specific gate months earlier.

Data lineage tracking belongs alongside observability rather than as an afterthought, because a validation team investigating a drifted model needs to know exactly which training data version produced it. Version-controlled dataset registries, timestamped retraining logs, and immutable records of which sensor or behavioral data contributed to a given model version let an engineer trace a classification error back to a specific data change rather than treating drift as an unexplained black box. Deng and Rattadilok's (2022) retraining cycle illustrates why this matters: without a recorded link between the expanded dataset and the resulting accuracy shift, a validation team downstream would have no way to distinguish a genuine model improvement from an artifact of how the new data was collected.

Particular weight falls on cybersecurity infrastructure given the empirical vulnerability landscape summarized in Table 2 below: hundreds of catalogued medical-device and software vulnerabilities, roughly half or more rated critical or high-severity, and an average exposure window measured in years rather than months before issues surface publicly (Bracciale et al., 2023; Mejía-Granda et al., 2024). Leaving automated dependency and vulnerability scanning out of the release gate leaves exactly this exposure window open for an ASD-focused application handling sensitive behavioral and biometric data. Zero-trust network segmentation, encrypted storage for sensor and behavioral data at rest, and continuous credential rotation extend the same logic from the validation pipeline itself into the production environment the pipeline is meant to protect.

A vulnerability scan that flags a finding accomplishes little without a defined response path. The infrastructure layer, therefore, pairs automated scanning with an incident-response workflow: a confirmed vulnerability above a defined severity threshold escalates to the same Human-in-the-Loop Governance layer used for low-confidence model outputs, routed to a security-cleared reviewer rather than an accessibility specialist, with a documented remediation timeline tracked against the exposure windows Bracciale et al. (2023) and Mejía-Granda et al. (2024) report. Treating security response as a governance-layer function, rather than a separate incident-management silo, keeps the audit trail of who reviewed what and when inside the same traceability infrastructure IEC 62304 already requires for other validation findings.

Table 2. Empirical findings from the approved reference set informing the framework's infrastructure and performance layers

Metric	Reported Value	Framework Layer Informed
Distinct vulnerabilities identified (national-scale device analysis)	661; >50% critical/high-severity	Infrastructure cybersecurity (§3.2)
Average deployment-to-disclosure exposure window	≈3.2 years	Infrastructure cybersecurity (§3.2)
Catalogued device/software vulnerabilities, 2001–2022	201 total; 106 critical for health	Infrastructure cybersecurity (§3.2)
Average healthcare data breach cost (2022)	$4.35 million	Discussion privacy stakes (§4)
Attention-detection accuracy, pre- vs. post-retraining	86.67% → 92.61%	Performance drift monitoring (§3.3)

Resilience testing, controlled failure injection, disaster-recovery drills, and network-partition simulation belong in this infrastructure layer rather than being treated as a separate operational concern. A validation framework that checks only model correctness while ignoring infrastructure failure modes leaves a caregiver-facing application vulnerable to outages precisely when a behavioral crisis makes reliability matter most.

3.3. Performance and Accessibility ValidationTypes of Graphics

Response latency and throughput are the performance metrics conventional software testing already knows how to check. Classification drift, confidence-score calibration, and the accuracy with which low-confidence outputs get routed to human review rather than acted on automatically are not, and this framework treats all of them as one performance surface rather than two. Deng and Rattadilok's (2022) sensory-management system shows why the drift half of that surface matters concretely: their attention-detection model's accuracy moved from 86.67% to 92.61% after a retraining cycle informed by additional labeled data, a change large enough that a validation pipeline unaware of the retraining event would be testing against a model materially different from the one it last certified.

Accuracy alone does not tell a validation team how much to trust a given prediction, which is why calibration assessment belongs in this performance surface as well. Reliability diagrams and calibration-error metrics standard evaluation tools in the broader machine learning literature (Goodfellow et al., 2016) reveal whether a model's stated confidence score actually tracks its real-world correctness rate; a poorly calibrated model might report high confidence on exactly the sensory-state classifications it gets wrong most often. Feeding calibration results into the human-in-the-loop governance layer's escalation thresholds, rather than relying on a fixed confidence cutoff chosen once at development time, keeps the routing decision anchored to how the model is actually behaving in deployment.

Reporting performance results in a way clinicians and accessibility specialists can actually use is itself a validation-engineering problem. A raw accuracy figure or calibration curve means little to the reviewers this framework's Human-in-the-Loop Governance layer routes findings to; translating a performance regression into a statement about which specific alert types or interaction sequences are affected and for which user segment is what turns a validation output into something a non-engineering reviewer can act on rather than simply defer back to the engineering team.

Generic screen-reader compliance checks are not what accessibility validation means here; treated as a first-class performance dimension rather than an afterthought, it requires test batteries specific to neurodiverse users. Predictable navigation patterns, minimized sensory load (motion, flashing content, abrupt audio changes), and consistent interaction sequences across sessions matter more for this population than they do for typical accessibility testing, which tends to prioritize screen-reader and keyboard-navigation compliance over sensory predictability. Usability instruments such as the System Usability Scale, used in Deng and Rattadilok's (2022) evaluation to compare ASD and typically-developing user groups mean scores of 70.5 and 68.3 respectively offer one quantitative anchor for this kind of validation, though the field lacks a standardized battery equivalent to the accessibility checklists available for conventional web software.

Age and developmental stage complicate this validation picture further. A sensory-classification model trained predominantly on younger children, as in Deng and Rattadilok's (2022) evaluation cohort, carries no guarantee of comparable accuracy for adolescents or adults with ASD, whose sensory presentation and communication patterns differ substantially. A performance validation plan that reports a single aggregate accuracy figure across an entire product's user base obscures exactly this kind of cohort-specific failure, which is why the accessibility and performance test batteries in this framework are specified as segmented by developmental cohort rather than pooled.

The absence of a standardized accessibility battery for neurodiverse users, noted above, has a practical workaround short of waiting for one to be developed: validation teams can define application-specific batteries now, documented with the same rigor as a formal standard, and treat cross-organization convergence on a shared battery as a longer-term goal rather than a precondition for validating today's deployed systems. Deng and Rattadilok's (2022) use of the System Usability Scale alongside a comparison group is one example of this workaround in practice: an existing general-purpose instrument repurposed with a comparison design specific enough to support an accessibility claim, rather than a purpose-built neurodiverse-accessibility instrument that does not yet exist.

Accessibility and performance therefore need to be checked together in regression testing: a model update that improves classification accuracy but pushes response latency past the threshold at which a sensory-overload alert becomes actionable has failed this framework's validation criteria even though it would pass a conventional accuracy-only test. AI-assisted anomaly detection across the CI/CV pipeline flagging performance regressions that a fixed threshold-based test might miss because the regression is gradual rather than a step change extends this drift-sensitive posture across releases rather than confining it to the release immediately following a retraining event.

4. Discussion: Broader Implications and Governance Risk

The framework proposed above carries implications that extend past the engineering practice it formalizes. Expanding validated AI-enabled ASD applications into rural and resource-constrained settings, where specialist behavioral therapy is least available, is one of the clearest potential benefits: a validation-engineering discipline that keeps these systems reliable as they scale is a precondition for that access benefit materializing rather than eroding as adoption grows. Sánchez de las Heras et al.'s (2026) observation that the field remains fragmented across disconnected pilot systems suggests this scaling has not yet happened at meaningful volume, and a shared validation discipline is one of the missing pieces standing between promising pilots and dependable deployment.

Trust operates as a second-order effect of validation quality rather than a separate design goal. Van Dort et al. (2026) found that clinicians' willingness to act on an AI recommendation tracks explanation quality more closely than raw model accuracy, which implies that a validation framework investing heavily in accuracy testing while treating explainability as a compliance checkbox will produce a system clinicians distrust regardless of how well it tests. The human-in-the-loop governance layer proposed in Section 3.1 is intended to close this gap by giving explainability findings to an architectural home rather than leaving them as narrative documentation attached to a model card no one consults during a clinical encounter. This structural emphasis on human oversight before an AI system acts on a low-confidence output also tracks the human-oversight and non-discrimination principles that transnational AI-ethics guidance has converged on (European Commission, High-Level Expert Group on AI, 2019), even though that guidance was not written with ASD-specific accessibility requirements in mind.

Bias risk is not a footnote. Training data for ASD-focused behavioral models are drawn disproportionately from clinical populations that do not reflect the full demographic and presentation diversity of ASD. A framework checking only aggregate accuracy will miss a model that performs well on average while failing systematically for an underrepresented subgroup. Section 3.3's segmented accessibility testing helps here, but only as far as a given deployment's validation team chooses to define its segments as a real limitation, not a solved problem.

Cost is an equity question this framework cannot resolve on its own. A four-layer validation architecture, continuous monitoring infrastructure, and a staffed human-in-the-loop review function represent a substantial engineering investment that a large health system can absorb far more easily than a small clinic or community-based ASD service provider. If rigorous validation becomes a de facto requirement only the best-resourced organizations can meet, the framework risks concentrating reliable AI-enabled ASD tools among patients who already have the most access to specialist care, the opposite of the access benefit described earlier in this discussion.

ASD applications handle sensory, behavioral, and often biometric data more sensitively than a typical consumer health app's. That exposure is not hypothetical: Bracciale et al. (2023) and Mejía-Granda et al. (2024) document it empirically, and healthcare breach costs averaging $4.35 million as of 2022 (Mejía-Granda et al., 2024) give it a quantified stake. This framework's cybersecurity layer treats that exposure as a continuous validation obligation, not a pre-audit penetration test run once and filed away.

Regulatory fragmentation compounds the cost question. A validation framework built primarily around FDA guidance (U.S. Food and Drug Administration, 2021) does not automatically satisfy the EU's parallel ethics and oversight expectations (European Commission, High-Level Expert Group on AI, 2019), and neither anticipates the accessibility-specific requirements this paper argues belong in the validation lifecycle. An organization deploying an ASD-focused application across multiple jurisdictions inherits the burden of reconciling these frameworks manually, since none of them yet speak to continuously learning, accessibility-sensitive healthcare software in the same terms.

Participatory design closes the discussion for good reason: involving clinicians, caregivers, and autistic individuals themselves throughout the validation process, rather than only at initial requirements gathering, prevents checks against the framework from becoming a purely technical exercise disconnected from the population it is meant to serve. The interoperability layer's connection to caregiver-facing systems is a partial structural answer, surfacing validation findings to the people who observe a system's real-world behavior most directly, though the framework does not by itself guarantee that this feedback loop translates into meaningful influence over validation criteria rather than one-way notification.

5. Guidelines for Graphics Preparation and Submission

Software validation engineering, as practiced against a fixed pre-release test surface, does not extend cleanly to AI-enabled ASD digital health applications whose behavior continues to evolve after deployment. This paper has proposed a framework that treats architecture, infrastructure, and performance as interlocking validation layers rather than sequential development phases, integrating continuous verification, explainability governance, accessibility testing, and cybersecurity monitoring into a single engineering lifecycle rather than distributing them across separate compliance processes that do not communicate with one another.

A synthesis, more than any single component, is the framework's contribution: existing standards (IEC 62304, ISO 29119, the FDA's AI/ML Action Plan) supply regulatory scaffolding without an operational model for continuous validation; the clinical-AI-governance literature supplies explainability and human-review practices without a software architecture to house them; and the AI-in-ASD-care literature supplies evidence that these systems work clinically without addressing how their engineering validation should function once deployed at scale.

Several limitations bound this contribution. The framework has not been implemented and tested against a production ASD-application deployment, so its architectural layers remain a design proposal rather than a validated engineering practice. Accessibility validation for neurodiverse users lacks the standardized test battery that exists for conventional web accessibility, leaving the framework's accessibility layer dependent on validation-team judgment rather than a settled checklist. Standardized accessibility batteries for neurodiverse users, empirical study of predetermined change-control plans in operational ASD-application deployments, and longitudinal tracking of model drift specifically in continuously learning ASD

behavioral models to establish what drift tolerance bands a CI/CV gate should actually enforce are the directions future validation-engineering research should pursue. Addressing these questions would move the framework proposed here from an architectural blueprint toward an evidence-based validation-engineering discipline for a population whose access to reliable digital health tools depends on that discipline existing. Cost and regulatory fragmentation, discussed in Section 4, further bound the framework's near-term applicability outside well-resourced healthcare organizations operating within a single regulatory jurisdiction.

Acknowledgements

The author received no dedicated funding, institutional support, or external assistance in the preparation of this manuscript. All conceptual development, analysis, and writing were carried out independently by the author.

Author contributions

Itendra Kumar Singh: Conceptualization, Methodology, Investigation, Writing – Original Draft Preparation, Writing – Reviewing and Editing, Visualization.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1]Ahuja, D., Sarkar, A., Chandra, S., & Kumar, P. (2022). Wearable technology for monitoring behavioral and physiological responses in children with autism spectrum disorder: A literature review.. Technology and Disability, 34(2), 69–84. https://openurl.ebsco.com/EPDB%3Agcd%3A4%3A20913700/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A157790411&crl=f&link_origin=www.google.com

[2]Amann, J., Blasimme, A., Vayena, E., Frey, D., & Madai, V. I. (2020). Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. BMC Medical Informatics and Decision Making, 20, 310. https://www.researchgate.net/publication/346542270_Explainability_for_artificial_intelligence_in_healthcare_a_multidisciplinary_perspective

[3]Boman, M. (2023). Human-curated validation of machine learning algorithms for health data. Digital Society, 2(3), 46. https://www.researchgate.net/publication/376121951_Human-Curated_Validation_of_Machine_Learning_Algorithms_for_Health_Data

[4]Bracciale, L., Loreti, P., & Bianchi, G. (2023). Cybersecurity vulnerability analysis of medical devices purchased by national health services. Scientific Reports, 13, 19509. https://www.nature.com/articles/s41598-023-45927-1

[5]Deng, L., & Rattadilok, P. (2022). A sensor and machine learning-based sensory management recommendation system for children with autism spectrum disorders. Sensors, 22(15), 5803. https://www.mdpi.com/1424-8220/22/15/5803

[6]European Commission, High-Level Expert Group on AI. (2019). Ethics Guidelines for Trustworthy Artificial Intelligence. https://ai.bsa.org/wp-content/uploads/2019/09/AIHLEG_EthicsGuidelinesforTrustworthyAI-ENpdf.pdf

[7]Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

IEC 62304:2006+A1:2015. Medical Device Software Software Life Cycle Processes.

IEEE Computer Society. Software Engineering Body of Knowledge (SWEBOK), Version 4.0.

ISO 13485:2016. Medical Devices Quality Management Systems.

ISO/IEC/IEEE 29119. Software Testing Standards.

[8]Liu, S., McCoy, A. B., Peterson, J. F., Lasko, T. A., Sittig, D. F., Nelson, S. D., Andrews, J., Patterson, L., Cobb, C. M., Mulherin, D., Morton, C. T., & Wright, A. (2024). Leveraging explainable artificial intelligence to optimize clinical decision support. Journal of the American Medical Informatics Association, 31(4), 968–974. https://pubmed.ncbi.nlm.nih.gov/38383050/

[9]Mejía-Granda, C. M., Fernández-Alemán, J. L., Carrillo-de-Gea, J. M., & García-Berná, J. A. (2024). Security vulnerabilities in healthcare: an analysis of medical devices and software. Medical & Biological Engineering & Computing, 62(1), 257–273. https://pubmed.ncbi.nlm.nih.gov/37789249/

[10]Pressman, R. S., & Maxim, B. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill.

Rajpurkar, P., Chen, E., Banerjee, O., & Topol, E. J. (2022). AI in health and medicine. Nature Medicine, 28, 31–38.

[11]Russell, S., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach (4th ed.). Pearson.

Sánchez de las Heras, J., Molina, A. I., & Lacave, C. (2026). [12]Application of artificial intelligence techniques for supporting people with ASD: a systematic literature review. Universal Access in the Information Society, 25(2), 65. https://www.researchgate.net/publication/403827655_Application_of_artificial_intelligence_techniques_for_supporting_people_with_ASD_a_systematic_literature_review

Sommerville, I. (2021). Software Engineering (11th ed.). Pearson.

[13]Spies, N. C., Farnsworth, C. W., Wheeler, S., & McCudden, C. R. (2024). Validating, implementing, and monitoring machine learning solutions in the clinical laboratory safely and effectively. Clinical Chemistry, 70(11), 1334–1343. https://academic.oup.com/clinchem/article/70/11/1334/7754656

[14]Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. Nature Medicine, 25(1), 44–56. https://www.nature.com/articles/s41591-018-0300-7

[15]U.S. Food and Drug Administration. (2021). Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device Action Plan. https://www.fda.gov/media/145022/download

[16]Van Dort, B. A., Engelsma, T., Sneekes, P., Peute, L., & Medlock, S. (2026). Explainable AI in hospital clinical decision support systems: A scoping review of healthcare professionals' perspectives. PLOS Digital Health, 5(5), e0001417. https://pubmed.ncbi.nlm.nih.gov/42133735/