NextGen Scientific Publication
American International Journal of Computer Science and Technology

Volume 4 Issue 2, Pg. No. 37-48, AIJCST-V4I2P105, 2022
https://doi.org/10.63282/3117-5481/AIJCST-V4I2P105

Original Article

Laboratory PCR Testing Systems and Regulated Software Practices During the COVID-19 Response

***Saket Mishra**

Associate Director, Thermo Fisher Scientific, South San Francisco.

Abstract:

The analysis focuses on the testing gap that emerged when public-health authorities needed rapid scale-up of sensitive molecular diagnostics while preserving traceability, safety, and regulatory compliance [1], [2]. The scope spans modular system design, laboratory automation, software configuration management, CI/CD for regulated diagnostic software, and secure data pipelines supporting end-to-end workflows. Key technical contributions highlighted include modular architecture patterns enabling scalable capacity and redundancy, laboratory automation strategies that reduce manual touchpoints, CI/CD approaches adapted to regulated software environments, and secure, auditable data pipelines that align with cybersecurity guidance [3]. Outcomes and implications are presented in anonymized form, emphasizing high-throughput capability improvements, verification and validation (V&V) success, and achievement of regulatory milestones under emergency and routine pathways [2], [4]. The review is relevant to public-health testing networks preparing for future emergencies and to organizations harmonizing software engineering practices with IVDR and FDA EUA expectations [2], [4]. Outcomes and implications are presented in anonymized form, emphasizing high-throughput capability improvements, verification and validation (V&V) success, and achievement of regulatory milestones under emergency and routine pathways [2], [4]. The review is relevant to public-health testing networks preparing for future emergencies and to organizations harmonizing software engineering practices with IVDR and FDA EUA expectations [2], [4].

Keywords:

Index Terms—Laboratory Automation, PCR Testing, Regulated Software, CI/CD, Quality Assurance, IVDR, FDA, Data Pipeline, Verification and Validation.

Article History:

Received: 10.02.2022

Revised: 25.02.2022

Accepted: 03.03.2022

Published: 19.03.2022

1. Introduction

Public-health authorities quickly recognised the central role of large-scale laboratory testing in COVID-19 response and real-time RT-PCR remains the reference standard for early diagnosis and surveillance [1], [5]. National and regional guidance highlighted the importance of sensitive assays, robust sample logistics, and secure reporting pathways to support case management and transmission controlciteWHO2020LabStrategy, CDC2020SpecimenGuidance. Laboratories were challenged to balance high daily specimen volumes with biosafety, workforce constraints, and evolving recommendations on sampling and interpretation [1], [6], [7]. The resulting environment required integrated solutions with instrumentation, automation and information systems with auditable software processes and reproducible configurations.

Copyright by the Author(s). This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International (CC BY-NC-ND 4.0) License (https://creativecommons.org/licenses/by-sa/4.0/)

The experience of national reference laboratory networks well illustrates the scale of this challenge. For example, in India, the Indian Council of Medical Research (ICMR) network grew from a handful of laboratories to several thousand testing sites during the pandemic and published operational analyses of testing timeliness and positivity trends in that network [5]. Such network-level expansions are emblematic of a broader trend observed across many countries: testing capacity needed to scale by orders of magnitude in a matter of months, while retaining the analytical and procedural rigour expected of a regulated diagnostic service [1], [5].

The main objective of this paper is to align the public and anonymised operational practices about system architecture, software and data infrastructure, software configuration management (SCM) and training, V&V strategies, regulatory documentation, CI/CD for regulated software, cross-functional integration, supply-chain resilience, and performance evaluation [1]–[3], [5]. Figure 1 illustrates a high-level view of the modular system architecture typically used to support such testing operations [1], [5]. The following sections elaborate each module and its interaction in more detail.

This review does not include proprietary assay performance data, vendor-specific implementation details, or confidential commercial agreements. Instead, it emphasises generic patterns, non-proprietary descriptions of workflows, and processoriented lessons that are transferable across technology stacks [1], [5]. The limitations are that performance metrics are anonymised, laboratory-specific contextual factors are not considered and the focus is on PCR-based workflows and not point-of-care rapid tests [1], [7]. The paper aims to identify reusable practices that can be incorporated into preparedness planning and regulatory strategies within these parameters. The rest of the paper is organised as follows. Section II contextualises the review within public-health advice, regulatory regimes, and peer-reviewed literature to support the ensuing discussion. The problem statement and scope are described in more detail in Section III. Sections IV to XII discuss system architecture, software and data infrastructure, SCM and training, QA and V&V, regulatory documentation, release management, cross-functional integration, supply-chain considerations, and performance evaluation, respectively. Section XIII concludes with limitations and future directions.

2. Related Work and Regulatory Landscape

This review draws on three complementary categories of source material: international and national public-health guidance, binding or harmonized regulatory frameworks for in vitro diagnostics, and peer-reviewed or standards-body literature describing operational and cybersecurity practice.

Table 1. Categories of Source Material Used in this Review

Category	Representative sources	Primary use in this review
Public-health guidance	WHO testing strategy and biosafety guidance [1], [7], [8]; CDC specimen guidance [6]	Workflow, biosafety, specimen handling
Regulatory frameworks	JAMA EUA landscape analysis [4]; EU IVDR [2]	Design controls, submission evidence
Standards and literature	NIST cybersecurity framework [3]; ICMR network analysis [5]	Security controls, regulatory trend context, throughput discussion

Table I summarizes how each category is used throughout the paper. For public-health guidance, the World Health Organization's interim laboratory testing strategy recommendations included risk-based prioritisation of testing, sample-type selection, and quality-assurance expectations for national laboratory networks during the early phase of the pandemic [1], [8]. A companion WHO interim guidance document outlined laboratory biosafety practices specific to coronavirus handling, which included biosafety-level recommendations, personal protective equipment (PPE) use, and waste-management considerations relevant to high-throughput laboratories [7]. The U.S. Centers for Disease Control and Prevention supplemented this with interim guidelines on the collection, handling, and testing of clinical specimens, addressing specimen types, transport conditions, and pre-analytical handling steps that significantly impact assay performance and downstream software workflows [6].

From a regulatory perspective, the U.S. Food and Drug Administration's Emergency Use Authorisation (EUA) framework provided a mechanism for in vitro diagnostic products to reach laboratories under an abbreviated, but still evidence-based, review pathway during the declared publichealth emergency. In the EU, Regulation (EU) 2017/746 on in vitro diagnostic medical devices (IVDR) set out a more comprehensive baseline of general safety and performance requirements, conformity assessment procedures and postmarket

surveillance obligations applicable when emergency derogations are no longer in force [2]. A peer-reviewed study published in JAMA looked at the wider EUSimilar to COVID-19 related medical products, emphasise the benefits of accelerated access and evidentiary trade-offs of emergency authorisation pathways [4].

Two additional sources underpin the operational and cybersecurity dimensions of the review, finally. The NIST Framework for Improving Critical Infrastructure Cybersecurity provides a widely adopted, sector-agnostic structure of Identify, Protect, Detect, Respond, and Recover functions that this review maps onto laboratory data-pipeline and SCM practices [3]. A peer-reviewed report of the ICMR laboratory network in India on testing timeliness and positivity trends provides a concrete, network-level example of how testing volume, turnaround time, and result quality were monitored and reported during a rapid scale-up [5], and is used in Section XII to motivate the discussion of anonymised throughput and reliability.

3. Problem Statement and Scope

Early in the pandemic, countries experienced substantial gaps between testing demand and available high-quality capacity, particularly for laboratory-based RT-PCR diagnostics [1], [5]. Shortages of consumables, limited automation, manual data handling, and fragmented software controls contributed to delays and inconsistent reporting in several settings [1], [5], [7]. Laboratories had to increase capacity while adhering to or surpassing test performance, biosafety, and regulatory requirements for in vitro diagnostics (IVDs) [1], [2]. This created a need for integrated system design, harmonised controls along the entire software lifecycle and clear evidence-generation practices to support emergency and routine approvals.

This paper is therefore limited to four axes: (i) system design and automation, (ii) software and data infrastructure, (iii) QA, V&V and test automation, and (iv) regulatory and release-management frameworks. The system design axis includes extraction modules, PCR instruments, liquid-handling automation, and laboratory information systems (LIS/LIMS) designed to enable high-throughput operation. The software axis covers instrument orchestration, data pipelines, security controls, interoperability interfaces, and SCM processes that provide traceability and reproducibility. The QA and V&V axis includes requirements traceability, test planning, automation frameworks and related documentation. The regulatory axis includes IVDR, FDA and emergency use pathways, release management with CI/CD, and audit preparedness [2], [4]. The paper does not attempt to review detailed clinical validation strategies or population-level epidemiologic uses of test data, which are widely covered elsewhere [5].

A further boundary of scope concerns the relationship between emergency and steady-state operation. Many of the practices described here were first adopted under EUA-style time pressure and were only later formalized into standing operating procedures aligned with IVDR-style design controls [2], [4]. The review therefore treats emergency-pathway practices not as a permanent substitute for routine regulatory rigor, but as an accelerated on-ramp that, if properly documented, can be converted into durable, auditable processes once the immediate crisis subsides.

4. System Architecture and Workflow

Typical high-throughput PCR testing systems consist of multiple layers of hardware and software components work-ing in concert from specimen receipt through result reporting [1], [6]. Typical core components include sample intake and accessioning stations, automated extraction instruments, liquid-handling robots for assay setup, PCR thermocyclers with fluorescence detection and storage devices for reagents and samples [1].

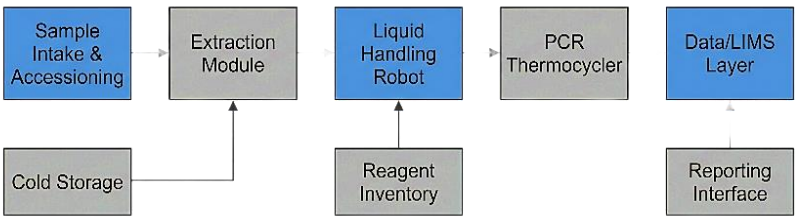

Figure 1. Conceptual High-Throughput PCR Testing Architecture, Including Intake, Extraction, Liquid Handling, Thermocyclers, and Data Systems

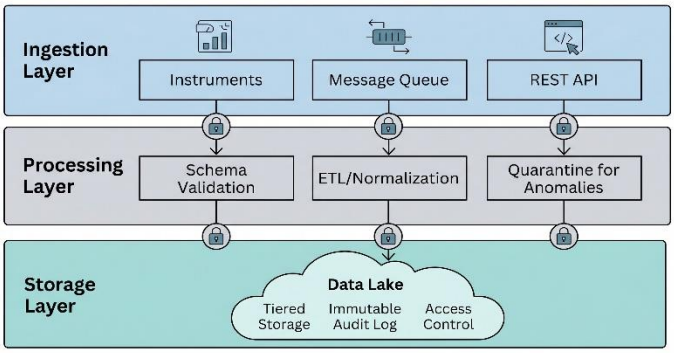

Figure 2. Generic End-To-End Workflow from Sample Receipt through Result Reporting, Showing Sample, Data, And Control Flows

At the software level, the key components are LIMS or LIS, instrument control software, middleware for orchestration and translation, and reporting interfaces to public-health systems. These components are summarised in Table II with typical functionality and associated personnel touchpoints [1], [5].

4.1. Modular Design and Scalability

The architecture is heavily based on modular design and scalability. Horizontal scaling is achieved by deploying multiple extraction and PCR modules in parallel, with orchestration software to assign worklists and manage contention for resources [1], [5]. Duplicate critical modules and interchangeable reagent kits validated with the same workflows create redundancy, reducing downtime in case of supply disruptions [7]. Physical layout also matters: centralized laboratories with large footprints can support highly automated lines, while distributed satellite labs reduce turnaround time in geographically dispersed regions but require robust logistics and remote monitoring [1], [5]. Figure 2 illustrates a generalized end-to-end workflow, including typical sample and data flows across modules [1], [6].

4.2. End-to-End Workflow

Most laboratories have a non-proprietary workflow consisting of specimen reception, accessioning with barcodes and entry into LIMS, and pre-analytical processing (e.g., aliquoting, inactivation if needed) [6], [7]. Nucleic acid eluates are then obtained via automated or semi-automated extraction.

Table 2. System Components and Typical Roles

Component	Typical Functionality	Primary Personnel Touchpoints
Sample intake	Barcoding, accessioning, initial QC	Accessioning staff, LIMS operator
Extraction modules	Automated nucleic acid extraction	Instrument operator, maintenance technician
Liquid handlers	Assay setup, plate transfers	Automation engineer, process engineer
PCR instruments	Amplification and detection	Instrument specialist, data reviewer
LIMS/orchestration	Tracking, result release, reporting	Lab manager, QA reviewer
Cold storage/freezer	Reagent and residual-sample retention	Inventory technician, cold-chain coordinator
Reporting interface	Electronic transmission to clinical/public-health systems	IT support, public-health liaison

These are transferred with liquid handlers or manually pipetted onto PCR plates with master mix and internal controls. The thermocyclers perform amplification and detection. Raw data are exported to analysis software that applies interpretation rules and quality checks. Results are then reviewed, approved and electronically transmitted to clinical systems and public health databases using

standard messaging formats [1], [6] Personnel touchpoints include accessioning staff, automation and instrument operators, QA reviewers, and IT support teams, each of which interacts with specific components, as shown in Table II.

4.3. Biosafety and Pre-Analytical Considerations

Because specimen handling happens before any softwaremediated step, biosafety practice has direct downstream consequences for data quality and software design. CDC interim guidelines specify acceptable specimen types, transport temperatures, and time-to-processing windows that, if exceeded, can degrade analytical sensitivity [6]. WHO biosafety guidance similarly recommends biosafety-level-2 (or enhanced BSL-2) practices for routine diagnostic handling of respiratory specimens, with additional precautions for procedures likely to generate aerosols [7]. These constraints influence the LIMS configuration decisions. For example, systems are typically configured to flag specimens that have exceeded a transporttime threshold or to require explicit operator acknowledgement before processing specimens collected under non-standard conditions so that any resulting uncertainty is recorded alongside the eventual result rather than discovered only after reporting.

5. Software and Data Infrastructure

Instrument orchestration software is responsible for coordinating worklists, sequencing commands to instruments, handling errors, and maintaining a consistent view of system state. Commands to instruments are designed to be as idempotent as possible, with explicit pre-conditions and post-conditions so that interrupted runs can be retried or resumed with minimal ambiguity. Timeouts, retry limits, and escalation paths are carefully defined to prevent cascading failures when a single module encounters an error. Logging and event correlation across instruments and middleware are critical for retrospective analysis, regulatory documentation, and incident investigation [1], [3].

5.1. Data Pipeline and Cloud Storage

Data pipelines connect instruments, LIMS, analytics systems and reporting targets over secure channels. Proposed models include authenticated ingestion via message queues or REST APIs, a landing zone for raw files, and ETL processes that normalise formats, validate schemas, and enrich data with metadata such as instrument identifiers and reagent lots [1], [6]. Final storage is usually cloud data lake architecture with tiered storage, immutable audit logs and fine-grained access controls, following general cybersecurity frameworks [3]. Often, data is serialised as structured text formats (e.g., CSV, JSON, XML) to enable validation and downstream integration. A generic data pipeline is shown in Figure 3 with ingestion, processing and storage layers [1], [3].

5.2. Interoperability and Interfaces

Interoperability is achieved through standard interface types such as RESTful APIs, HL7-style messages, and file-based interfaces for legacy systems [6]. Data validation routines impose schema constraints, integrity checksums, and business rules before records are accepted into authoritative stores. ETL jobs have rejection and quarantine flows for anomalous records so they can be investigated without polluting production datasets. Security controls include encryption in transit (for example, TLS), encryption at rest using managed keys, and role-based access control (RBAC) aligned with least-privilege principles. Access and administrative actions are logged to provide evidence for audits and regulatory submissions [1], [3]

5.3. Mapping Security Controls to a Cybersecurity Framework

Many laboratory IT teams find it useful to organise controls according to the five core functions of a widely used cybersecurity framework: Identify, Protect, Detect, Respond and Recover [3]. Organising controls in this way may help keep security investment systematic rather than ad hoc. The Identify activities include inventories of instruments, servers and software assets, and data-flow diagrams similar to those summarised in Figure 3. Protect activities include the encryption, RBAC, and least-privilege measures described above, as well as the operating systems and middleware patch management. Activities for Detect involve log aggregation and anomaly detection at the instrument, middleware, and network levels. Respond activities define incident-handling runbooks, e.g., who gets notified when an anomaly is detected and what containment steps are taken. Backup and restore procedures for LIMS databases and configuration are part of the recovery activities, together with post-incident review processes that feed lessons learned back into the Identify and Protect functions. Framing laboratory cybersecurity in these five functions also simplifies subsequent regulatory mapping, since reviewers under both EUA and IVDR pathways increasingly expect evidence that a structured cybersecurity process, rather than a one-time control list, is in place [2], [3].

6. Software Configuration Management and Training

In regulated diagnostic environments, SCM is the foundation for reproducible and auditable software behaviour. Recommended repository practices include adopting either a trunk-based development model with short-lived feature branches or a Gitflow-style model with clearly defined release branches and long-term support streams. Semantic versioning is a formal way to express compatibility expectations. The major, minor, and patch components correspond to breaking changes, backward - compatible functionality, and defect fixes. Tags and signed releases in the version control system serve as anchors for traceability in the V&V documentation and regulatory submissions [2], [3].

6.1. The Complexity of Managing Heterogeneous SOUP Components

Managing configuration artifacts and third-party components, including software of unknown provenance (SOUP), is one of the more difficult and persistent challenges in multi-module diagnostic systems, and deserves particular emphasis. For example, one orchestration application may have direct or transitive dependencies on dozens of open-source libraries, vendor-supplied instrument drivers, operating-system packages, and embedded firmware images, each with its own release cadence, support lifetime, and vulnerability disclosure process. In practice, this is complicated by a number of factors:

- Heterogeneous provenance. Components originate from public open-source registries, commercial vendors, and internally maintained forks, each with different levels of documentation and different willingness to disclose known issues.
- Version sprawl. Without active governance, the same nominal library can end up at different pinned versions across instrument-control, middleware, and reporting modules, complicating both vulnerability remediation and behavioral consistency.
- Transitive dependency depth. Modern build tooling readily pulls in indirect dependencies several levels deep, many of which are never directly inspected by engineering staff.
- Long-lived regulated baselines. When a software version has been validated and submitted as part of an emergency or routine regulatory filing, any subsequent upgrade to one component demands a documented impact assessment, creating incentives to postpone the update despite vulnerability disclosures [2], [3].

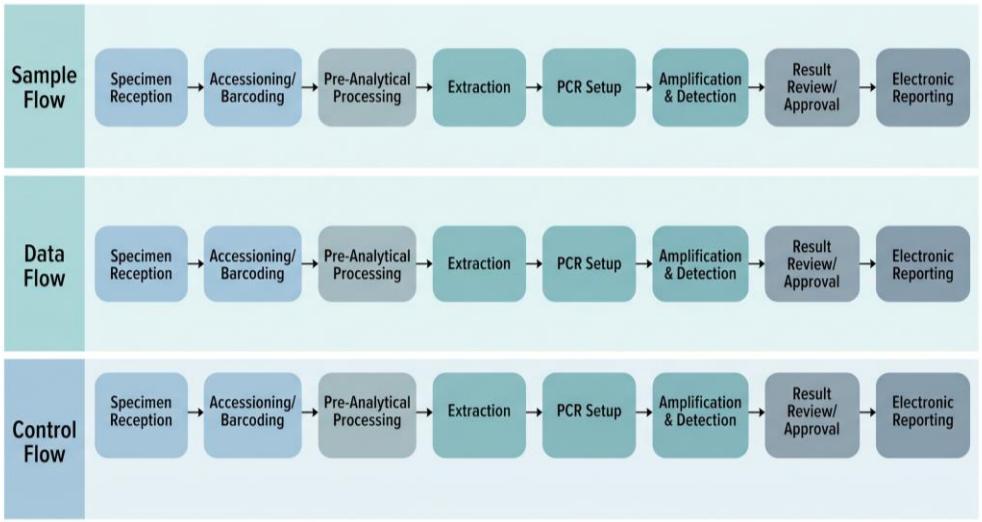

Figure 3. Representative Secure Data Pipeline for PCR Testing, Including Ingestion, Validation, ETL, and Cloud Data Lake Storage

The best and most efficient approach seen to manage this complexity, while minimising both security and validation errors, is to treat the software bill of materials (SBOM) as a living, automatically generated artefact rather than a manually maintained document. A complete SBOM lists open-source and commercial dependencies, versions and licensing terms; this SBOM is automatically updated during builds where possible [3]. Generating the SBOM as a build-pipeline step (rather than retrospectively) means the inventory will always reflect what was actually compiled and shipped, not what was intended. The combination of generating SBOMs and automated vulnerability scanning on each build enables new issues to be triaged against the exact set of components in use, which dramatically reduces false negatives (missed vulnerable components) and false positives (flagging components that were already retired). A risk-tiering policy that classifies each SOUP component by exposure – for example, components parsing untrusted network input versus

purely internal utility libraries – further focuses limited remediation effort where it has the greatest safety impact. Finally, requiring a lightweight, explicit risk assessment for each new addition or version bump, not just major upgrades, ensures the SBOM remains up to date without adding burdensome process overhead to daily development. Table III summarises minimal control methods versus recommended controlled practices for SCM and SOUP management.

Reproducible build containers and infrastructure-as-code declarations further reduce configuration drift between development, test, and production environments, which further reduces the number of SOUP-related discrepancies that only show up at deployment time. Training is essential: teams benefit from structured sessions on Git workflows, branching strategies, code review criteria, and documentation of SCM conventions, as well as targeted training on how to read and act on SBOM and vulnerability-scan output rather than treating it as an opaque compliance artifact. Consistently applied over time, these practices can yield measurable improvements in the form of reduced merge conflicts, better traceability from requirement to deployment, and faster and controlled rollback in the event of a defect

Table 3. SCM and Soup Management Practices

Area	Minimal control approach	Recommended regulated approach
Third-party components	Ad hoc updates, limited tracking	SBOM, vulnerability scanning, provenance records
Configuration files	Local copies, manual backups	Versioned artifacts in SCM, signed releases
Branching	Long-lived unreviewed branches	Trunk-based or structured Gitflow with reviews
Build reproducibility	Manual environment setup	Containerized builds with artifact repository
SOUP risk classification	Treated uniformly, reviewed only at major upgrades	Risk-tiered by exposure, reviewed at every version change

7. Quality Assurance, Test Automation And V&V

V&V strategies for diagnostic software should be consistent with the overall device risk classification and regulatory expectations [2], [4]. A requirements traceability matrix records high-level requirements, detailed design elements and test cases, creating a transparent link from clinical or operational requirement to implemented functionality and evidence. Test planning also includes defining acceptance criteria, environmental conditions, and negative and boundary scenarios that represent realistic misuse or edge cases. Standardised test protocol and report templates help to ensure consistency and risk-based testing ensures that the verification effort is proportionately greater for higher risk functionality.

7.1. Test Automation Frameworks

As systems evolve, test automation frameworks become ever more critical to upholding quality. Selenium WebDriver can automate web UI regression testing of LIMS portals, dashboards and administrative tools, while RESTful API automation frameworks validate the service behaviour with a wide variety of payloads and error conditions [6]. In a common implementation, Selenium WebDriver scripts drive browser-level interactions against a staging LIMS instance, exercising accessioning, result-release, and report-generation screens with both legitimate and purposefully malformed data inputs, while a separate RESTful API automation suite issues authenticated HTTP requests directly against the orchestration and reporting services, evaluating response payloads, status codes, and latency against defined acceptance thresholds. Running the UI and API suites in parallel, rather than purely sequentially, allows defects originating in the API layer to be quickly distinguished from those specific to UI rendering or browser behaviour. Data-driven approaches, where test logic is separated from input data, support efficient coverage of variant scenarios and languages. Robotic Process Automation (RPA) may be applied to cross-system validations, such as reconciling results between instrument outputs and LIMS records, or simulating end-user workflows across application boundaries. Figure 4 illustrates a layered testing stack, from unit and integration tests to system-level automated checks [3].

7.2. Outcome Metrics

Outcome metrics give insight into the effectiveness of QA and V&V without exposing proprietary performance values. Other common measures are requirement coverage by automated tests, distribution of defect severities detected before and after release, and median time from code change to detection of regression in the CI pipeline . By analysing trends over time, we can identify systemic issues, for example, specific modules that are generating recurring defects or specific integration points that are showing instability.

Resource efficiency gains from automation are reflected in the percentage of regression suites run without manual intervention, and the ability of teams to reallocate manual testing time to exploratory and scenario-based testing.

8. Regulatory And Compliance Documentation

Central to IVDR and FDA regulatory frameworks for IVDs are design controls and documentation [2], [4]. Core nonproprietary artefacts include system and software requirements specifications, architecture descriptions, detailed design documents, risk analyses and hazard logs, test plans and protocols, verification and validation reports, release notes, and change control records. These collectively are evidence that requirements were fully implemented, risks were identified and mitigated and changes were controlled. Cybersecurity documentation such as threat models and mitigation strategies has also become increasingly important for connected diagnostic systems [3].

8.1. IVDR and FDA QA/RA Frameworks

The IVDR and FDA frameworks are similar in their focus on demonstrable design control, but are different in structure and emphasis. IVDR requires manufacturers to prove compliance with general safety and performance requirements, implement a quality management system controlling software in the context of the overall device life cycle and produce a performance evaluation report supported by analytical and clinical performance data, if applicable [2]. The FDA's framework includes both a standard premarket pathway with a comparable design history file and an EUA pathway with an abridged but structured evidence package for a declared public-health emergency [4]. An accelerated authorisation pathway for COVID-19 products, part of a broader EUA landscape, has been reviewed by a peer-reviewed analysis which found it greatly improved access to new diagnostics and therapies for patients and laboratories, but also observed that the evidentiary basis underlying individual EUAs varied considerably, underscoring the importance of strong internal QA/RA discipline even when the external regulatory bar is temporarily lowered [4].

8.2. Emergency Use Authorization Access Frameworks

Emergency authorisation access frameworks, for example EUAs for in vitro diagnostics in the United States, or similar national derogations in Europe, require clear documentation of intended use, assay principles and limitations [4]. A large number of in vitro diagnostic products reached laboratories via the EUA pathway during 2020 and 2021, a volume that the JAMA analysis attributes to the urgency of the public health response and regulators' willingness to accept a shortened, but still structured, evidence package [4]. For IVDR filings, manufacturers must provide evidence of conformity with the general safety and performance requirements, submit performance evaluation reports and have in place a quality management system that covers software processes throughout the device lifecycle [2]. Table IV compares the two pathways on several practically relevant dimensions.

8.3. How This Review Supports Regulatory Submissions

Mapping software artifacts to regulatory expectations involves demonstrating how requirements and hazards are linked to specific tests and mitigations, how software versions used in validation match the release versions, and how configuration and SOUP are controlled across the lifecycle. The framework described in this review helps assemble such evidence systematically, even when systems are being developed under time pressure, in three concrete ways. First, the SCM and SOUP practices in Section VI produce the version-pinned, vulnerability-scanned inventory that both EUA reviewers and IVDR auditors increasingly expect to see referenced in a cybersecurity section of the submission [2], [3].

Table 4. Illustrative Comparison of Emergency and Routine Regulatory Pathways

Dimension	Emergency Pathway (e.g., FDA EUA) [4]	Routine Pathway (e.g., EU IVDR) [2]
Trigger	Declared public-health emergency	Standard market access, any time
Evidence depth	Abbreviated analytical/clinical summary	Full performance evaluation report
QMS expectation	Still required, often streamlined review	Full QMS conformity assessment
Duration of validity	Tied to emergency declaration	Indefinite, subject to post-market surveillance
Typical timeline	Weeks	Months to over a year

Second, the requirements-traceability and test-automation practices in Section VII generate the verification evidence that maps directly onto to the performance evaluation report structure required under IVDR and onto the analytical performance summaries expected under an EUA [2], [4]. Third, the CI/CD release-gating and change-control practices in Section IX produce the signed, versioned

build artifacts and change-request records that regulators and auditors use to confirm that the software actually validated is the software actually deployed. Taken together, these practices reduce the marginal effort required to convert day-to-day engineering records into a coherent submission package, whether the immediate goal is an emergency authorization or a full IVDR conformity assessment.

8.4. Audit Preparedness

Good documentation and artefact organization also helps with audit preparedness for ISO style assessments where auditors would expect to see regularity in SOPs, training records and easy access to design and test artefacts that are mapped to device versions. Maintaining a single, version-controlled index that cross-references each regulatory artifact (requirements, hazard log entries, test reports, SBOMs, and release records) to the corresponding software release tag is a comparatively low-cost practice that substantially reduces the time needed to respond to auditor information requests, since reviewers can be pointed directly to the relevant evidence rather than having staff reconstruct the linkage during the audit itself.

9. Release Management and CI/CD

CI/CD pipelines for regulated diagnostic software must find the right balance between agility and control. Staged workflows can be implemented using systems like Jenkins Pipeline where source code is checked out, static analysis is performed, containerised builds are executed, and automated tests are run before any artefact is considered for promotion [3]. Each pipeline stage produces evidence: code quality reports, test result summaries, and signed build artifacts with associated metadata. The pipeline permits promotion to the staging or production environments only if predefined quality gates are met and relevant checklists are completed.

9.1. Illustrative Jenkins Pipeline Structure

For this domain a representative Jenkins declarative pipeline is structured into clearly separated stages that are independently auditable. The Checkout stage retrieves the exact source revision under build, recording the commit hash in pipeline metadata.

Table 5. CI/CD Stages and Regulatory Controls

Pipeline stage	Typical automation steps	Regulatory control examples
Build	Dependency resolution, static analysis	Signed artifacts, SBOM capture
Test	Unit, integration, UI, API tests	Test evidence archived, traceability links
Staging	Deploy to staging environment	Reproducible configurations, approval gates
Release	Artifact promotion, tagging	Change control records, release approval checklist
Post-release	Monitoring, hotfix pipelines	Patch validation records, communication logs

The Static Analysis stage runs linters and security scanners (e.g., the SBOM and vulnerability-scanning steps described in Section VI) and fails the build if new highseverity findings are introduced. The Build stage constructs the application within a versioned container image, making the build environment itself reproducible and traceable to a particular image tag. Test stage runs unit, integration, API and UI automation suites discussed in Section VII, and archives structured test reports as build artefacts. The Quality Gate stage checks aggregate pass/fail state, coverage thresholds, and any open high-severity vulnerabilities, blocking promotion if any gate fails. The Artifact Signing stage cryptographically signs the build artifact and attaches the corresponding SBOM, satisfying provenance requirements. Finally, the Promotion stage deploys the signed artifact to staging or production only after the relevant approval checklist, described in the following subsection, has been completed. Table V summarizes typical pipeline stages and associated regulatory controls.

9.2. Change Control and Release Governance

Change control and release governance are closely coupled with CI/CD. Change requests reference specific issues, requirements, or risks and must include impact analyses, proposed mitigations, and testing implications. Approval workflows involve appropriate QA/RA representatives, and release notes capture the scope and justification of each release. Rollback planning includes preparation of tested rollback artifacts, documented procedures, and conditions under which rollback is preferred over forward fixes.

9.3. Post-Release Support

Post-release support processes define triage protocols for incidents, severity classification, investigation workflows, and communication channels to affected users, ensuring that any issues with potential patient impact receive immediate attention and that remedial measures are documented for future audits [2], [4]. Patch validation for hotfixes follows an abbreviated but still evidence-

producing version of the full pipeline described above, so that even urgent changes retain a documented chain of static analysis, automated testing, and sign-off before deployment.

10. Integration with Cross-Functional Teams

For effective development, cross-functional teams must be integrated, because in diagnostic platforms, assay design, hardware, firmware, and software typically evolve in parallel. Biology teams define targets, primer/probe sets, and assay conditions; hardware teams tune instrument mechanics and thermal performance; firmware teams implement low-level control; and software teams manage orchestration, data handling, and reporting. Such separation of concerns is required for specialised teams to work efficiently, but it also means that each module is, by design, developed and validated largely independently of the others. Seamless integration across these independently developed modules is therefore not a peripheral concern, but a critical control point in its own right: a perfectly correct software release, an individually wellcharacterized assay, and a properly calibrated instrument can still combine to produce an erroneous result if the interfaces and assumptions between them are not jointly verified. Without structured coordination, misalignments between these domains can produce defects that are difficult to detect and rectify late in the lifecycle, precisely because each team's own testing may show no fault within its own module.

10.1. Cross-Discipline Coordination Mechanisms

Cross-discipline coordination mechanisms include regular integration meetings, shared milestone tracking dashboards, jointly owned system-level acceptance criteria defined by biology, hardware, firmware, and software leads. Figure 5 shows an example coordination model, emphasizing the central role of integration leads and shared backlogs. Misalignment is visible only at the system level, so these coordination mechanisms are deliberately weighted toward shared, end-to-end artefacts, such as a single interface-contract document or a shared defect backlog visible to all four disciplines, rather than separate per-discipline tracking that must later be reconciled by hand.

10.2. System-Level Integration Testing

Common test datasets and reference scenarios that exercise the whole pipeline from sample intake to result reporting, including failure modes, are beneficial for system-level integration testing. The environment staging process deploys representative hardware and software configurations prior to release, allowing for end-to-end verification within an environment that closely resembles production. Because each module's own unit and component tests cannot, by construction, exercise the interfaces between modules, system-level integration testing is treated as a distinct and mandatory verification activity rather than an optional extension of unit testing, with its own acceptance criteria and sign-off requirements separate from those of the individual disciplines.

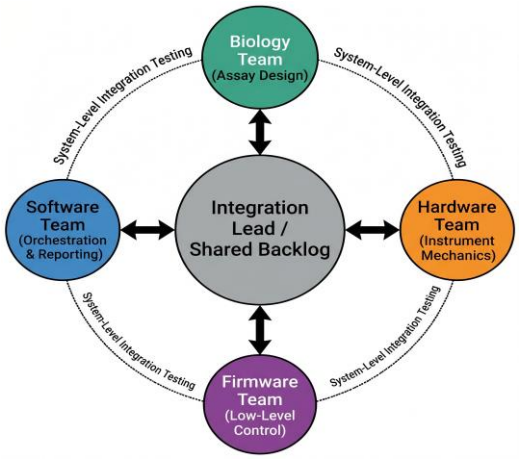

Figure 5. Illustrative Cross-Functional Integration Model Spanning Biology, Hardware, Firmware, and Software Teams with Shared Milestones

10.3. Issue Management and Escalation

Issue management and escalation processes include one point of contact per domain, formal ticketing systems for capturing issues and decisions, and predefined escalation trees for high-pressure incidents such as field failures or regulatory queries. A clear escalation

path is particularly important for cross-functional defects, since an issue that appears at the software layer may in fact originate in an assay reagent lot or a firmware timing change; routing such issues first through a single point of contact per domain, rather than directly between individual engineers, helps ensure that the eventual root-cause investigation is informed by all relevant disciplines from the outset rather than being narrowed prematurely to the domain where the symptom was first observed.

11.. Supply-Chain and Logistics Considerations

Forecasting reagents and consumables for PCR testing for COVID-19 was a forecasting exercise in high uncertainty. Extraction kits, PCR reagents, plastics and personal protective equipment were affected by global supply chain disruptions [1], [7]. Guidance to the public on prioritising testing for some populations over others when supplies were limited informed laboratory planning scenarios [1], [7]. Forecasting methods used case trends, historical use and contingency assumptions, often with conservative buffer stock policies. Alternate suppliers were assessed and pre-qualified where possible, with bridging validations and risk assessments documenting equivalency. Inventory systems incorporated supplier lead times, shelf-life and cold-chain requirements into procurement algorithms.

11.1. Consumable Validation and Interchangeability

Consumable qualification and interchangeability required strong protocols. Each new lot was tested for positive and negative controls to ensure that it met the pre-defined acceptance criteria before use in routine testing [1], [5]. When other consumables (e.g. different plate types or tips) were considered, laboratories performed bridging studies on representative sample sets, documented any observed changes in performance and modified workflows or interpretation rules accordingly. Tracking trends in positivity and turnaround time can provide early signals that a consumable substitution or supply disruption is impacting the quality of results across multiple sites rather than a single laboratory. This can be done through central monitoring of network-level indicators, such as the ICMR laboratory network [5].

11.2. Inventory and Procurement Workflows

Inventory and procurement workflows increasingly relied on electronic systems that incorporated usage data from LIMS and automation logs, offering near real-time visibility into consumption and remaining stock. Automated alerts at reorder thresholds and exceptions for items with especially long lead times or tight cold-chain constraints improved resilience during subsequent surges. Where laboratories were part of a larger network that aggregated procurement signals across sites, and not just at the individual-laboratory level, this allowed for early identification of region-wide shortages, and supported coordinated rather than purely local mitigation [5].

12. Performance Evaluation and Verification Results

The functional verification of PCR test systems involves tests at different levels, from single software modules to system-level workflows involving instruments and human operators [2]. Functional testing verifies that the system accepts valid inputs and provides expected output under normal conditions. Robustness tests change environmental conditions, data formats, or common operator errors to test the graceful detection and handling of deviations. Stress tests examine how things perform when you have high throughput or resource contention. The theory is that you want performance degradation to be predictable and manageable, not just suddenly fail.

12.1. Throughput and Reliability Assessment

Methods to assess throughput and reliability need to respect confidentiality but still provide meaningful insights [1], [5]. Labs may report throughput as qualitative bands (e.g., low, moderate, high) mapped to broad ranges rather than specific numbers, and reliability as qualitative uptime categories and typical recovery patterns. Published network-level analyses such as the ICMR laboratory network study of testing timeliness and positivity [5] demonstrate that reporting of turnaround-time and positivity trends can be operationally meaningful for public-health decision-making without revealing site-specific proprietary performance figures.

12.2. Sources of Variability

Variability can be due to sample quality, operator technique and environmental factors such as temperature and humidity. Some methods to measure impact include designed experiments where factors are varied systematically, inter-operator comparison studies, and statistical process control charts. And when exact numbers aren't shared, visualisations and stories can tell a story about how variation was measured and how processes were adjusted to keep performance stable over time.

13. Limitations And Future Directions

The primary limitations of this review are that the review discusses lab-based PCR workflows and does not elaborate on the fast-changing world of point-of-care and at-home testing, while many software and regulatory principles are still relevant [1]. In terms of future technical opportunities, deeper integration of automation across pre-analytical, analytical and postanalytical phases, incorporation of sequencing workflows for variant detection and genomic surveillance, and expansion of analytics capabilities for operational optimisation, anomaly detection and predictive maintenance [5] are some of the opportunities. The policy and preparedness implications point to the need for sustained investment in modular laboratory infrastructure, harmonised data standards, and rapid regulatory pathways that maintain evidentiary rigour in emergencies [1]– [4]. Embedding regulated software engineering practices such as robust SCM, secure CI/CD, and comprehensive V&V into routine operations can reduce response time and increase reliability when new threats arise.

References

[1] World Health Organization, "Laboratory testing strategy recommendations for covid-19: Interim guidance," https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/ laboratory-guidance/, 2020.

[2] European Commission, "Regulation (eu) 2017/746 of the european parliament and of the council on in vitro diagnostic medical devices (IVDR)," https://eur-lex.europa.eu/legal-content/EN/TXT/?uri= CELEX:32017R0746, 2017.

[3] National Institute of Standards and Technology, "Framework for improving critical infrastructure cybersecurity, version 1.1," https://www.nist.gov/cyberframework, 2018.

[4] B. N. Rome, J. Avorn, and Others, "Emergency use authorizations of covid-19–related medical products," JAMA, vol. 326, no. 16, pp. 1607– 1608, 2021. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/ PMC8689422/

[5] K. Sethuraman and Others, "Covid-19 testing, timeliness and positivity from ICMR's laboratory network in India," Indian Journal of Medical Research, vol. 154, no. 5, pp. 620–628, 2021. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC8641892/

[6] Centers for Disease Control and Prevention, "Interim guidelines for collecting, handling, and testing clinical specimens from persons for coronavirus disease 2019 (covid-19)," https://stacks.cdc.gov/view/cdc/85935, 2020.

[7] World Health Organization, "Laboratory biosafety guidance related to coronavirus disease (covid-19): Interim guidance," https://www.who.int/emergencies/diseases/novel-coronavirus-2019/ technical-guidance/laboratory-guidance, 2020.

[8] ——, "Laboratory testing strategy recommendations for covid19: Interim guidance," https://research.rug.nl/en/publications/ laboratory-testing-strategy-recommendations-for-covid-19-interim-, 2020.