Paper ET24

R We There Yet? A Readiness-Assessment Framework for Adoption of R and Open-Source Solutions

Narasimhan (Narasi) Kaliyamoorthy, Syneos Health®, Austin, USA

ABSTRACT

How ready is your organization to use R and open-source solutions for programming and regulatory submissions? Can readiness be assessed systematically and reliably? While enthusiasm and pilot successes are growing, the lack of a standard assessment framework necessitates that organizations define their own workflows, tools, and strategies for adoption. This variability raises critical questions: How can we assure regulators of consistency and compliance? How reproducible are results across environments? Who is accountable for governance? True readiness requires a structured assessment to enable broad, consistent, and confident industry adoption that is sustainable in the long term. This paper proposes a reference framework for readiness assessment across four dimensions: People (talent, training, leadership), Infrastructure (computing environment, security, reproducibility), Process (workflows, validation, quality), and Governance (roles, compliance, change, and risk management). Moving beyond ad-hoc or fear-of-missing-out adoption, this framework enables a systematic, informed, and sustainable approach to adopting R and open-source solutions.

INTRODUCTION

The pharmaceutical industry is undergoing a paradigm shift as it transitions from the long-standing dominance of proprietary software like SAS toward a more flexible, open-source landscape powered by R and Python. This evolution is fueled by a shifting talent pool of modern data scientists, the need for advanced methodologies to handle complex trial designs, and a regulatory environment that prioritizes reproducibility over specific software brands.

While industry leaders have already successfully navigated end-to-end open-source submissions, many organizations are still stalled by concerns over validation and governance. Bridging this gap requires moving beyond experimental phases through structured organizational maturity assessments, transforming the adoption of open-source tools from a perceived risk into a calculated strategic evolution.

INDUSTRY AT CROSSROADS

The pharmaceutical industry is currently at a high-stakes crossroads with clinical and statistical programming foundation. For decades, the sector relied on a closed-source monopoly, primarily using proprietary software like SAS. This was not just a software choice; it was a cultural standard driven by the need for validated systems that could survive the stringent scrutiny of regulators like the FDA.

Today, the tide has turned toward open-source languages like R and Python. This shift is a survival response to the complexity of modern clinical data and the use of diverse data sources such as Real-World Evidence (RWE) that legacy systems struggle to handle with agility. It is not just about efficiency and potential cost savings, but also about innovation and tapping into global next generation talent. When a new statistical method or machine learning model is developed in academia, it is often released as an open-source package very quickly. In the legacy model, a programmer might wait years for a proprietary vendor to implement that same innovation. This has made open-source adoption no longer just a nice-to-have, but a strategic necessity.

MOVING BEYOND PILOTS

The industry is currently wrestling with how to support the rigorous validation and reproducibility required by regulators like the FDA while using code that is frequently being updated by the community for features and bug fixes. To address this, and efficiently moving beyond successful pilots, there is currently a massive rise in pre-competitive collaboration, with rival companies putting their efforts together to build shared, open-source libraries like Pharmaverse that everyone can trust and reuse.

THE REGULATORY EVOLUTION

Traditionally, the pharmaceutical industry has relied on Computer System Validation (CSV), a method that uses heavy documentation and repetitive certification to preserve system environments and reproducibility. While this works for older software with infrequent changes, it is too broad for modern open-source tools like R, which update often. Recently, regulatory guidance has shifted toward Computer Software Assurance (CSA), which focuses core validation on high-risk areas while allowing simpler checks for lower-risk tasks.

Regulatory agencies are now actively co-designing the infrastructure for open-source adoption and support risk-based approaches like CSA. The shift from a SAS-only world to a multi-language ecosystem (R, Python, Julia, etc.) is driven by own digital modernization goals of regulators like the FDA. Reviewers now routinely use R and Python internally to double-check sponsor results. The FDA, for example, is encouraging the use of modern Statistical Computing Environments (SCEs) and containerized submissions that meet regulations like 21 CFR Part 11, regardless of the language used.

APPROACHES TO ADOPTION

The pharmaceutical industry is currently deploying three primary strategic approaches to transition from legacy proprietary systems to open-source solutions in clinical trials.

THE HYBRID APPROACH

A conservative parallel processing entry point approach that involves using R or Python alongside SAS to cross-verify results. While it ensures high confidence and satisfies traditional validation requirements, it is resource-intensive as it requires dual-programming capabilities of workforce.

THE MODULAR APPROACH

A modular approach that uses open-source tools for specific high-value tasks that proprietary software handle poorly, such as interactive Shiny dashboards for monitoring, complex statistical modeling, or advanced visualizations. This accelerates innovation without disrupting the core legacy infrastructure. However, it introduces significant complexities and fragmentation in the workflow.

THE COLLABORATIVE APPROACH

A collaborative approach that embraces industry standards, led by initiatives like the Pharmaverse, moving away from guarded internal macros to co-developing shared, open-source R packages like {admiral}. This open-first strategy reduces the individual validation burden and enables a standardized infrastructure that regulatory agencies like the FDA already recognize.

The Collaborative Approach is highly recommended because it replaces redundant internal development with shared, industry-vetted standards like the Pharmaverse. While it offers unparalleled efficiency and regulatory alignment by spreading the validation burden across the industry, it demands a significant cultural shift and requires careful management of long-term governance for community-maintained code.

BARRIERS TO ADOPTION

Successful adoption of open-source solutions is primarily hindered by the validation burden, as the responsibility for proving software reliability under GxP and 21 CFR Part 11 shifts from the vendor to the sponsor. Organizations must also navigate supply chain risks, specifically the potential for "orphaned code" if community developers cease updates, and a workforce talent gap of technical fluency (in R, Python, etc.). Cultural inertia also remains a barrier as moving away from guarding proprietary macros to contributing to shared industry standards like the Pharmaverse requires a fundamental shift in how sponsors perceive their competitive advantage.

THE HIDDEN COSTS OF “FREE”

It is a common misconception that open-source translates to free-of-charge in a regulated environment. While the initial licensing fee is zero, the Total Cost of Ownership (TCO) of open-source solutions can often rival or exceed legacy proprietary costs. In the SAS world, you pay for a validated black box. In the open-source world, you are the vendor. Extensive man-hours are needed for GxP compliance and risk assessments. Significant investment is needed for Infrastructure, such as containerized environments and package managers. High hiring and training costs are needed for specialized skillsets that are currently in high demand and command higher salaries than traditional legacy programmers. Rigorous legal oversight required for open-source solutions to mitigate security and licensing risks incur added costs to the sponsors.

THE VALIDATION FALLACY

Another common misconception is that open-source code is inherently secure and reliable simply by virtue of public scrutiny. While transparency enhances security, peers often assume others have already vetted the code. This creates a dangerous cognitive bias where community popularity metrics, such as download counts and peer ratings are mistaken for verified technical accuracy. These metrics are unreliable, inadmissible, and insufficient for GxP compliance within the stringent regulatory environment of the pharmaceutical industry.

THE SWISS ARMY KNIFE TRAP

In the transition to open-source, organizations often fall into the trap of designing tools or packages to perform too many distinct "Swiss Army Knife" functions. This leads to excessive complexity and reduced maintainability. Attempting to force R or Python to behave exactly like SAS, and recreating massive, complex code libraries ignores the modular, functional programming paradigm of the open-source ecosystem. The result is a clumsy replica of the legacy system that accumulates technical debt, quickly leading to integration fatigue, without delivering the benefits of modern open-source solutions.

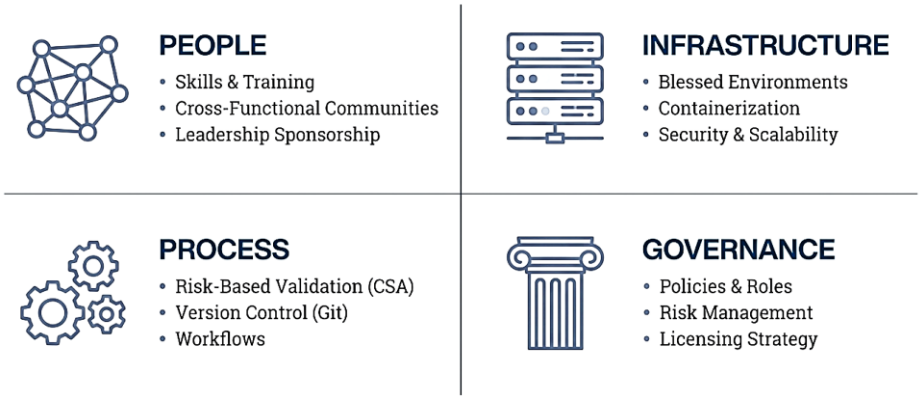
BRIDGING THE GAP: ASSESSMENT FRAMEWORK FOR ADOPTION READINESS

Based on successful industry case studies on open-source adoption, end-to-end submission workflows, regulatory guidance, and software engineering principles, this paper proposes a structured framework for assessing organizational readiness for open-source solutions, preparing for adoption and execution of adoption strategies.

This framework evaluates readiness across four distinct but interconnected dimensions:
People, **Infrastructure**, **Process**, and **Governance**.

Successfully integrating R and open-source tools into clinical trials is a strategic shift rather than a technical one. Success depends on organizational maturity across all four dimensions: **People** (training and hiring a workforce that aligns with open-source mindset); **Infrastructure** (setting up stable, immutable computing environments); **Process** (software quality assurance and risk-based validation); and **Governance** (oversight and clear accountability). This structured approach provides a clear roadmap for adoption while avoiding common pitfalls.

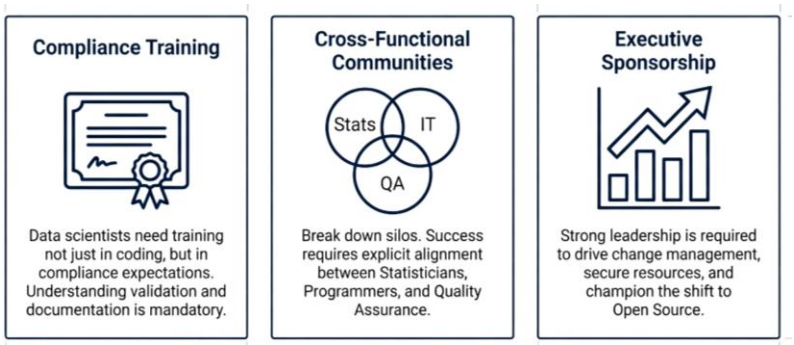
DIMENSIONS OF READINESS ASSESSMENT



DIMENSION 1: PEOPLE - CULTIVATING TALENT AND CULTURE

The most significant barrier to open-source adoption is often cultural, not technical. The transition from SAS to R represents a fundamental shift in how analysts interact with data, moving from a procedural, macro-based workflow to a functional, object-oriented paradigm.

Readiness in the People dimension requires more than just training for syntax. It demands a cultural shift toward open-source based development. Unlike SAS, which handles many data management tasks implicitly, R and Python require users to be aware of data types, memory management, and package dependencies.



RECOMMENDATIONS

- Adopt a hybrid training model, progressively upskilling with low-risk tasks before moving to high-risk tasks.
- Successful adoption requires executive sponsorship that understands the strategic value of open-source solutions, beyond mere cost savings on licenses. Leadership must champion the cultural shift and protect teams from the productivity dip that inevitably occurs during the initial learning curve.
- A consistent finding in organizations with successful adoption is the emergence of the package steward role. This individual is not necessarily a dedicated IT professional but a senior member responsible for the curation, risk assessment, and maintenance of specific open-source packages used by the teams. This distributes the burden of governance and ensures that subject matter experts (SMEs) engage in the validation process.
- Adopting open-source solutions requires breaking down silos. In the proprietary model, knowledge is often centralized in specific macro libraries owned by a few experts. In the open-source model, knowledge is distributed. Readiness involves building internal groups that operate like open-source communities, sharing code, conducting peer reviews, and collectively maintaining internal packages.

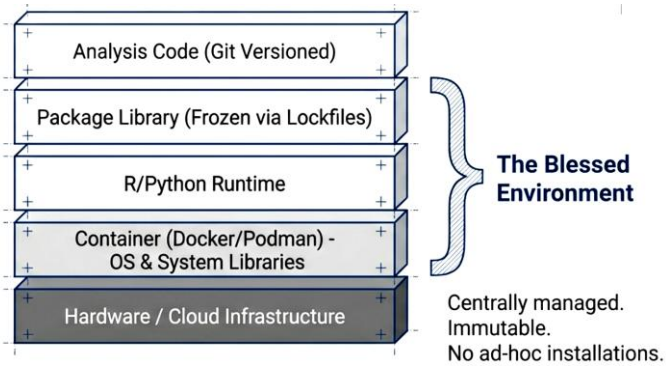
KEY READINESS QUESTIONS (PEOPLE)

- Does leadership visibly champion and fund open-source initiatives as a strategic capability rather than a cost-cutting measure?
- Is there a formal plan for upskilling teams, not just in syntax, but in modern practices (Git, CI/CD)?
- Are roles clearly defined for the new ecosystem (e.g., who validates a package vs. who uses it)?
- Is there a formal structure for internal communities of practice to facilitate knowledge sharing?

DIMENSION 2: INFRASTRUCTURE - THE FOUNDATION OF REPRODUCIBILITY

In a regulatory setting, the infrastructure's ability to produce a result is secondary to the ability to reproduce that result exactly the same way, potentially multiple years later. Open-source solutions, with their rapid update cycles and complex dependency trees, pose a unique challenge to infrastructure and reproducibility.

Organizations must move away from ad-hoc local installations and use managed environments where package stewards control the standardized repositories. This ensures that every team is working with consistent set of tools and approaches.



RECOMMENDATIONS

- Containerization and "Blessed" Environments: Use containers (e.g., Docker, Podman) to freeze the entire environment (OS, R/Python versions, and libraries). This is a reliable way to guarantee the long-term reproducibility required by the regulatory agencies.
- Centralized Package Management: Deploy a private repository to host a white-listed snapshot of community packages. This mitigates supply chain risks and ensures every team uses identical, risk-assessed code versions.
- Git-Centric Workflow: Integrate version control (GitHub/GitLab) directly into the environment. This provides the mandatory traceability (who, what, when) required by 21 CFR Part 11 for clinical submissions.

KEY READINESS QUESTIONS (INFRASTRUCTURE)

- Can a regulator precisely reproduce results using the environment provided, even 5 or 10 years later?
- How do we manage and document package dependencies for a specific submission?
- Is the computing environment immutable and centrally managed, preventing ad-hoc user or system updates that break reproducibility?
- Are our audit trails sufficient to reconstruct an analysis from code to results?

DIMENSION 3: PROCESS - FROM VALIDATION TO ASSURANCE

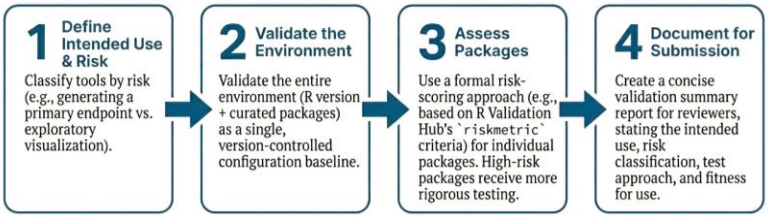
The Process dimension addresses how work is done, documented, and verified throughout the open-source infrastructure. To ensure a successful transition to open-source, the workflows must adopt automated, risk-based procedures that align with modern software development life cycles.

The traditional Computer System Validation (CSV) model, characterized by extensive documentation and scripted testing of every feature, is ill-suited for the vast open-source ecosystem. The move toward Computer Software Assurance (CSA) encourages a risk-based approach supported by regulators:

- High Risk (Primary Efficacy/Safety): Requires rigorous, scripted testing and distinct validation evidence. This applies to packages used for primary endpoint analysis.
- Medium/Low Risk (Exploratory/Visualization): Can rely on unscripted testing, ad-hoc verification, and community usage metrics. This applies to tools used for data exploration or internal reporting.

Process readiness involves explicit verification of the algorithms critical to the analysis using a tiered risk-based package validation strategy using tools like {riskmetric} from the R Validation Hub:

- Tier 1 (Core): Base R and recommended packages. Validated via installation qualification (IQ).
- Tier 2 (Established): High-quality community packages (e.g., dplyr, ggplot2). Assessed via quality metrics (test coverage, maintainer activity, errors, and correction cycles).
- Tier 3 (Niche/Internal): Packages with lower usage or internal development. Require full functional testing and code review.



RECOMMENDATIONS

- Use Validated Standard Packages: The Pharmaverse is a collaborative effort to build a suite of open-source R packages specifically for clinical reporting (e.g., {admiral} for ADaM derivation, {teal} for interactive exploration). Adopting these tools standardizes the process, allowing companies to share the burden of development and validation.
- Collaborative Code Review: Institutionalize a Git-based peer-review process. Instead of a separate validation programmer rewriting the code, use pull requests to document the review, discussion, and approval of code, creating a built-in audit trail for GxP compliance.
- Standardized Contribution Rules: Set up a clear process for how internal teams can contribute to or update shared libraries (like the Pharmaverse). This includes defining who can whitelist a new package and how bug fixes are reported and tracked.
- Agile Documentation: Move away from static validation reports toward living documentation. Use tools like Quarto or R Markdown to integrate code, output, and validation evidence into a single, reproducible document that updates automatically as the code changes.

KEY READINESS QUESTIONS (PROCESS)

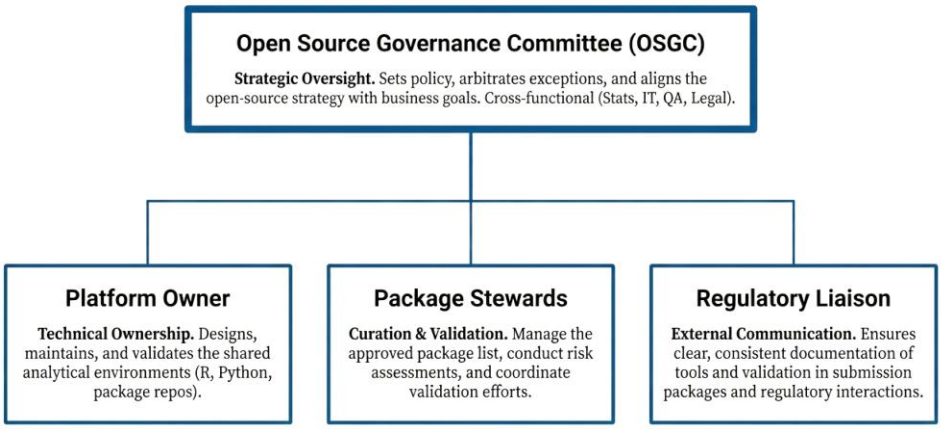
- Is our package validation approach documented, risk-based, and aligned with GxP principles?
- Is our coding practices integrated with version control from the start?
- How do we prove that our R-based outputs are CDISC-compliant?
- Is there a defined process for selecting, vetting, and promoting R packages into the blessed environment?

DIMENSION 4: GOVERNANCE - ACCOUNTABILITY AND STRATEGY

Governance provides guardrails that allow innovation to occur safely. Without clear governance, open-source adoption becomes chaotic and exposes the organization to legal and operational risks.

Organizations should set up an Open-Source Governance Committee (OSGC) including representatives from Statistics, Programming, IT, QA, and Legal. This body acts as the central authority for:

- Defining the "Blessed" environments and updating schedules.
- Approving new packages for the white-list.
- Arbitrating exceptions for high-risk tools.
- Managing the lifecycle of open-source tools and overseeing deprecated packages.



A unique challenge of using open-source code is license compliance and intellectual property risk. While MIT and Apache licenses are generally permissive and safe for corporate use, "copyleft" licenses (like GPL/AGPL) can pose risks if code is modified and distributed. Governance policies must explicitly define acceptable license types to prevent intellectual property contamination. The key regulatory expectations include:

- Demonstrable Fitness for Purpose: The sponsor must provide objective evidence to prove the software is validated for its specific intended use in the clinical or quality process.
- Absolute Control and Traceability: The sponsor must maintain complete control over the software's build, changes, and deployment. Every aspect of its architecture, risk analysis, and history must be documented to prove clear accountability.
- Independent Lifecycle Management: The sponsor must prove the agile ability to support the software over its entire lifecycle, including patching critical cybersecurity vulnerabilities, without being restricted by a third party's terms or timeline.

RECOMMENDATIONS

- Risk-Based Package Governance: Implement a formal white-listing process. Instead of treating all packages equally, categorize them by risk (e.g., Core/Stable vs. Experimental) and define clear criteria for when a community-developed package is mature enough for GxP use.

- Lifecycle Management and Orphan Planning: Define a policy for long-term maintenance. This includes monitoring the health of open-source communities and setting up a Plan B (internal maintenance or commercial support) for critical packages that might lose their original maintainers.
- Inner-Sourcing and Contribution Policy: Set up rules of engagement for how employees contribute back to the community. This governs Intellectual Property (IP) protection, ensuring that general tools are shared while proprietary clinical algorithms stay protected.
- Audit-Ready Traceability Standards: Standardize how metadata is captured. Governance must mandate that every submission package includes a precise record of the language version, package versions, and author credentials to satisfy 21 CFR Part 11 compliance.
- Regulators Care About Control, Not Licenses: Regulators are not license experts. They are focused on whether a company can prove it has total, documented control over its software's validation, security, and maintenance for its entire lifecycle.
- The Risk is in the Conflict of Obligations: A license becomes a regulatory problem when its legal requirements (like mandatory source code disclosure) undermine a company's ability to meet its regulatory obligations (like protecting proprietary, safety-critical code and managing cybersecurity).

KEY READINESS QUESTIONS (GOVERNANCE)

- Who is accountable for our open-source strategy and who arbitrates exceptions?
- Do we have a formal process for approving new packages and managing their lifecycle?
- Is there a documented policy for acceptable software licenses?
- Does the organization have an Open-Source Governance Committee to manage accountability and risks?

COST OF ADOPTION VS. COST OF INACTION

While the Cost of open-source license fees is zero, the total cost of ownership includes infrastructure setup, training, and governance overhead. These costs can be substantial in the short term. However, the Cost of Inaction is arguably higher. By staying hooked to the legacy systems, organizations lose the ability to leverage modern methods such as machine learning and advanced capabilities that are native to the R and Python ecosystems. The risk of falling behind the industry's evolution outweighs the operational friction of the adoption and transition:

- Financial: Continued reliance on escalating legacy license fees for proprietary software.
- Operational: Inefficiency due to manual workflows and lack of automation.
- Strategic: The inability to recruit top talent is a critical risk. Data scientists want to work with modern tools; forcing them to use legacy systems leads to high turnover and difficulty in hiring. Also, the inability to use modern tools and statistical designs that are native to open-source solutions places the organization at a competitive disadvantage.

The safest and most effective approach is not to avoid open-source solutions, but to manage it intentionally. A clear governance, combined with smart infrastructure, compliant processes and effective people and skills management allows companies to leverage the power of open-source solutions, while ensuring that they stay in full compliance with regulatory expectations.

EXECUTION STRATEGY: THE 90-DAY ROADMAP

To move from fear-of-missing-out or ad-hoc adoption of open-source, to a structured, strategic adoption, a proposed 90-day roadmap derived from successful case studies is as below:

- Days 1-30: Form the Open-Source Governance Committee (OSGC) - assemble a cross-functional team made up of Statistics, Programming, IT, QA, and Legal. Define the charter and get executive sponsorship.
- Days 31-60: Draft Policy. Define the traffic light rules for packages (Green=Pre-approved, Yellow=Review, Red=Banned). Establish license policies and compliance guardrails.
- Days 61-90: Freeze Environment. Create the first Blessed snapshot of the computing environment. Validate the installation.
- Next Step: Pilot. Select a low-risk study (e.g., exploratory, or retrospective analysis) to evaluate the end-to-end workflow using the new infrastructure. Use a hybrid or collaborative approach to scale up.

CONCLUSION

The pharmaceutical industry has been asking, "Is R ready for Pharma?" The evidence, from successful NDAs, FDA pilots, and robust validation frameworks suggests the answer is a definitive "Yes." The more pertinent question is, "Is Pharma ready for R?"

- Successful adoption requires a fundamental shift toward modularity and embracing the open-source mindset. This ensures that the open-source solutions stay agile, scalable, and easier to validate. By following this roadmap, we can transform open-source solutions from a perceived risk into a compliant, sustainable engine for innovation.
- Readiness is not a binary state but a spectrum of maturity across People, Infrastructure, Process, and Governance. Organizations that treat open-source adoption merely as a technical swap for SAS will struggle with compliance and stability.
- Conversely, those that use this transition as a catalyst to modernize their validation frameworks (CSA), embrace reproducibility (containers), and a collaborative engineering culture will realize significant gains in efficiency, talent retention, and analytical capability.
- With successful end-to-end R-based submissions and pragmatic hybrid R/SAS based submissions, the roadmap is becoming clearer every day. The most advanced organizations are no longer debating if R can be used but are instead focusing on building robust internal frameworks to use it systematically and safely. Instead of asking "Can we use R?", organizations should evaluate the more important question: "How do we get ready to use R systematically?"
- We are there. The tools exist, the regulatory pathway is open, and the community is vibrant. The onus is now on organizational leadership to adopt the structured frameworks necessary to walk that path with confidence. By systematically assessing readiness across the four dimensions proposed in this paper, organizations can move from ad-hoc piloting to sustainable, enterprise-scale adoption of open-source solutions. These four dimensions are not separate silos; they are interlocking components of a single readiness strategy.

REFERENCES

1. Computer Software Assurance for Production and Quality System Software; Guidance for Industry, FDA, 2026. https://www.fda.gov/media/188844/download
2. Roche's first end-to-end R journey to submission, 2025. https://phuse.s3.eu-central-1.amazonaws.com/Archive/2025/Connect/US/Orlando/PRE_SA01.pdf
3. J&J: A Hybrid SAS/R Submission Story - R Consortium, 2025. https://r-consortium.org/webinars/jnj-hybrid-sas-r-submission-story.html
4. R in FDA Submissions: Lessons Learned from 5 FDA Pilots - Appsilon, 2025. https://www.appsilon.com/post/r-in-fda-submissions
5. Risk Assessment for R Package Validation - Atorus Research, 2025. https://www.atorusresearch.com/r-package-risk-assessment-why-a-hybrid-approach-is-the-future-of-package-validation/
6. Learnings from the First R-Based Submission to FDA by Novo Nordisk - Appsilon, 2024. https://www.appsilon.com/post/first-r-based-submission-to-fda-by-novo-nordisk
7. Risk Metrics to Evaluating R Packages, riskmetric, 2024. https://pharmar.github.io/riskmetric/
8. A Guide to R Package Validation in Pharma - R-bloggers, 2024. https://www.r-bloggers.com/2024/10/a-guide-to-r-package-validation-in-pharma/
9. Using R in a Regulatory Environment: some FDA perspectives, 2018. https://www.washstat.org/presentations/20181024/Schuette.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Narasimhan (Narasi) Kaliyamoorthy, Associate Director, FSP Biometrics
Company: Syneos Health®
Email: n_kaliyamoorthy@sloan.mit.edu

DISCLAIMER

The opinions expressed in this paper are solely those of the author and do not necessarily represent the opinions, recommendations, or practices of any organization. All brand and product names are trademarks of their respective companies.