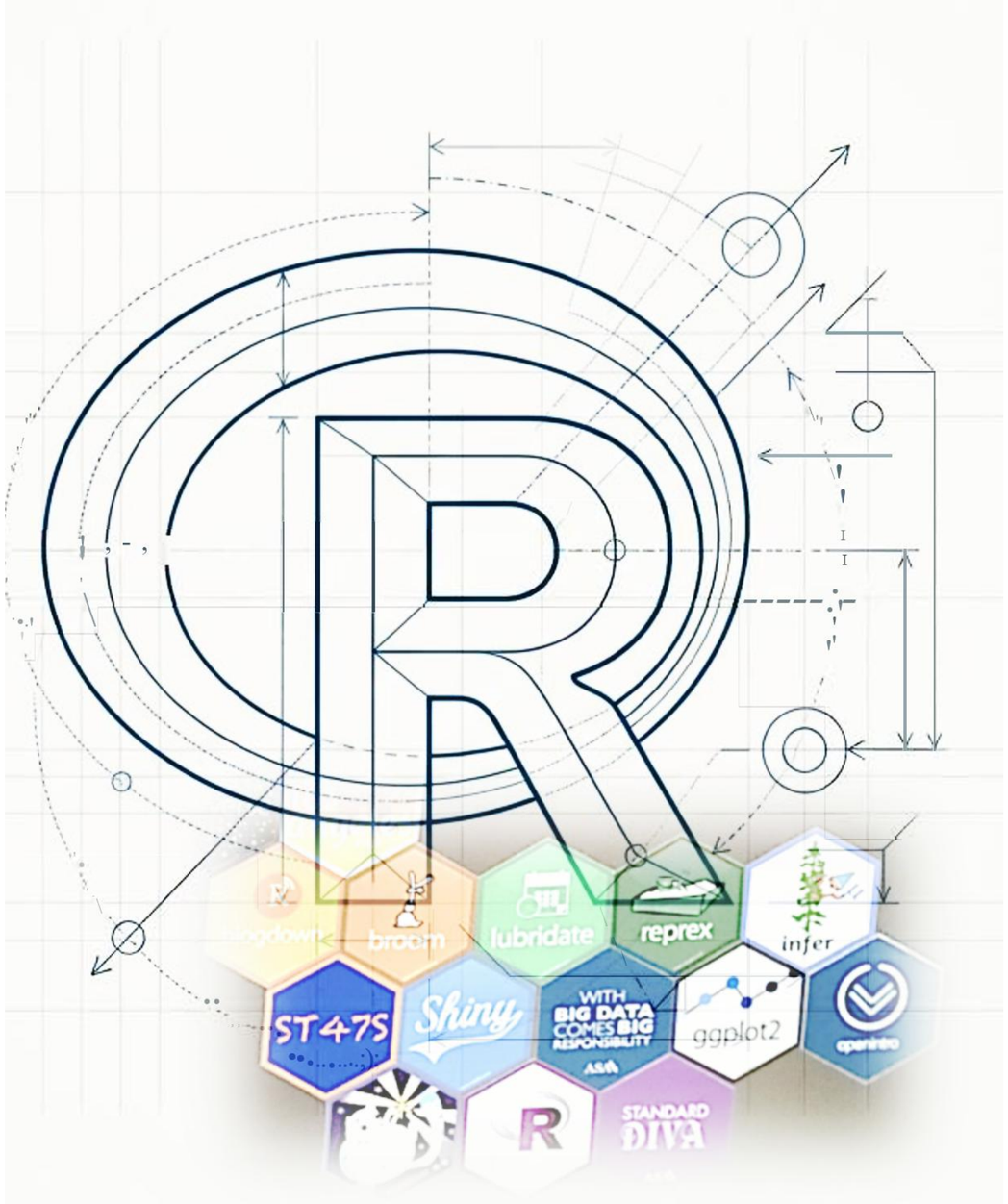
R We There Yet?

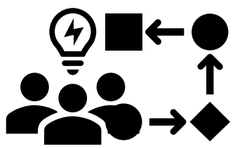
A Readiness-Assessment Framework for Adoption of R and Open-Source Solutions

Roadmap for moving from ad-hoc to confident, compliant and sustainable adoption

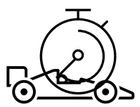


[PHUSE 2026 SDE | San Marcos, TX | Jan'26 | Narasi Kaliyamoorthy]

The Industry is at a Crossroads: Embracing Innovation While Ensuring Compliance



The Opportunity



Accelerate analysis and shorten timelines.

Access a vast ecosystem of powerful, cutting-edge tools and standards.

Attract and retain top analytical talent.

Increase flexibility and analytical capabilities.

The Challenge

Inconsistent workflows raise reproducibility risks.

Unclear governance creates accountability gaps.

Lack of a standard validation approach leads to duplicated effort.

Persistent regulatory anxiety hinders confident submissions.

Current State of Open-Source Adoption: Where R We?

Proven Precedents: Regulatory Pathways Are Open

Novo Nordisk (First R-Based NDA)

Successfully submitted an R-based NDA. FDA requested code with minimal dependencies.

Key Lesson
Reproducibility is paramount. Use containers.

Roche (End-to-End R)

Submitted NDA to FDA, EMA, and NMPA using the pharmaverse ecosystem.

Key Lesson
Collaboration works. Engage regulators early.

J&J (The Hybrid Model)

Used SAS for ADaM derivations and R for TLFs.

Key Lesson
Pragmatism. Hybrid migration reduces risk.

R Consortium Pilot 4

Submitted interactive Shiny apps via WebAssembly.

Key Lesson
Technical Innovation. FDA is open to modern delivery.

The Question Has Changed

~~Is R ready for Pharma?~~

Is Pharma ready for R?

Challenges to Adoption: Trading the Hum of Stability For the Sonic Boom

SAS/Legacy	R and Open-Source
Centralized Change Management	Distributed Governance
Stable and Time Tested	New and Evolving
Proprietary	Peer Reviewed
Hum of Stability	Breaking the Sound Barrier

Open-source adoption is a massive transformation of mindset and culture, not just a change in programming language.

LEGACY STABILITY

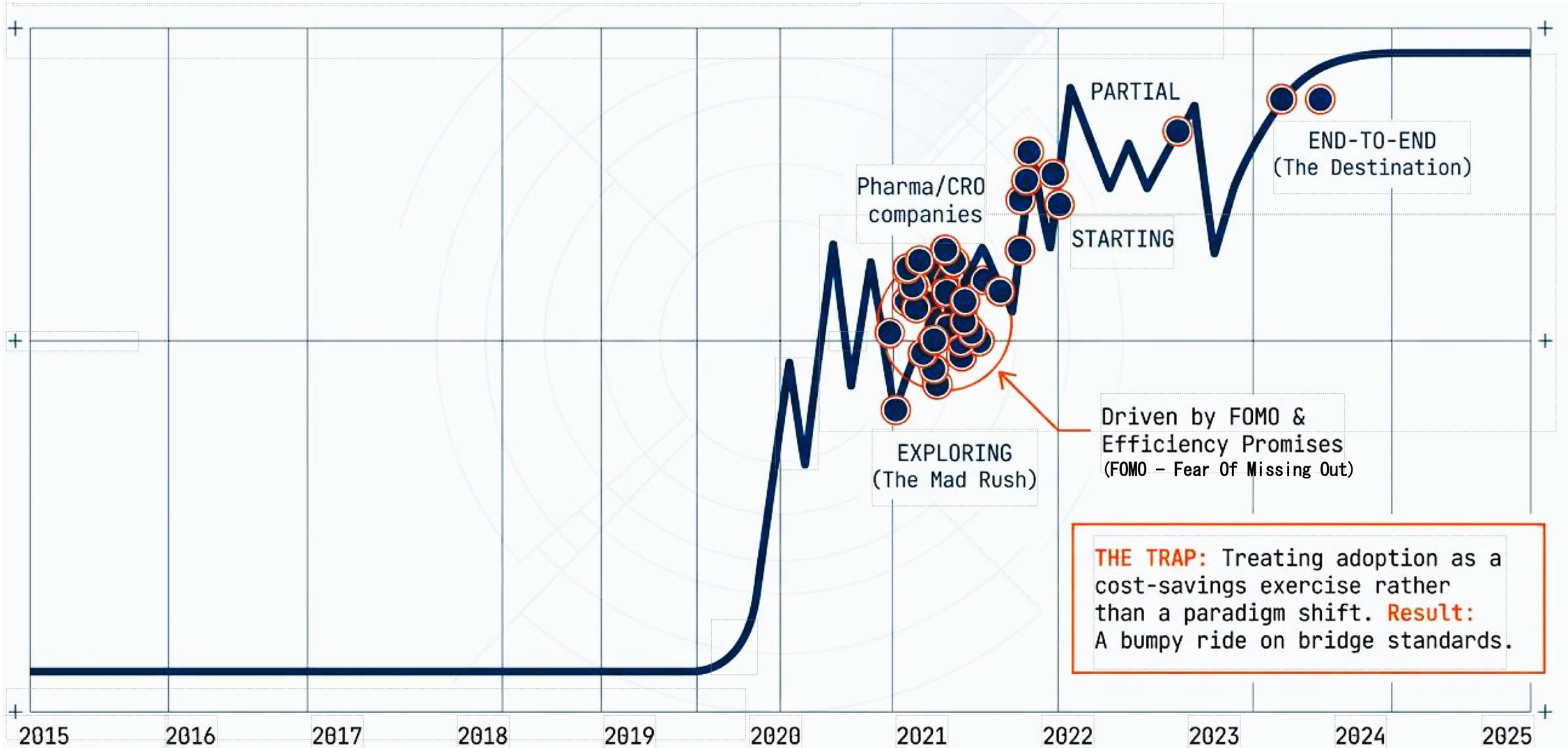
Time-tested stability of a 747

OPEN-SOURCE EVOLUTION

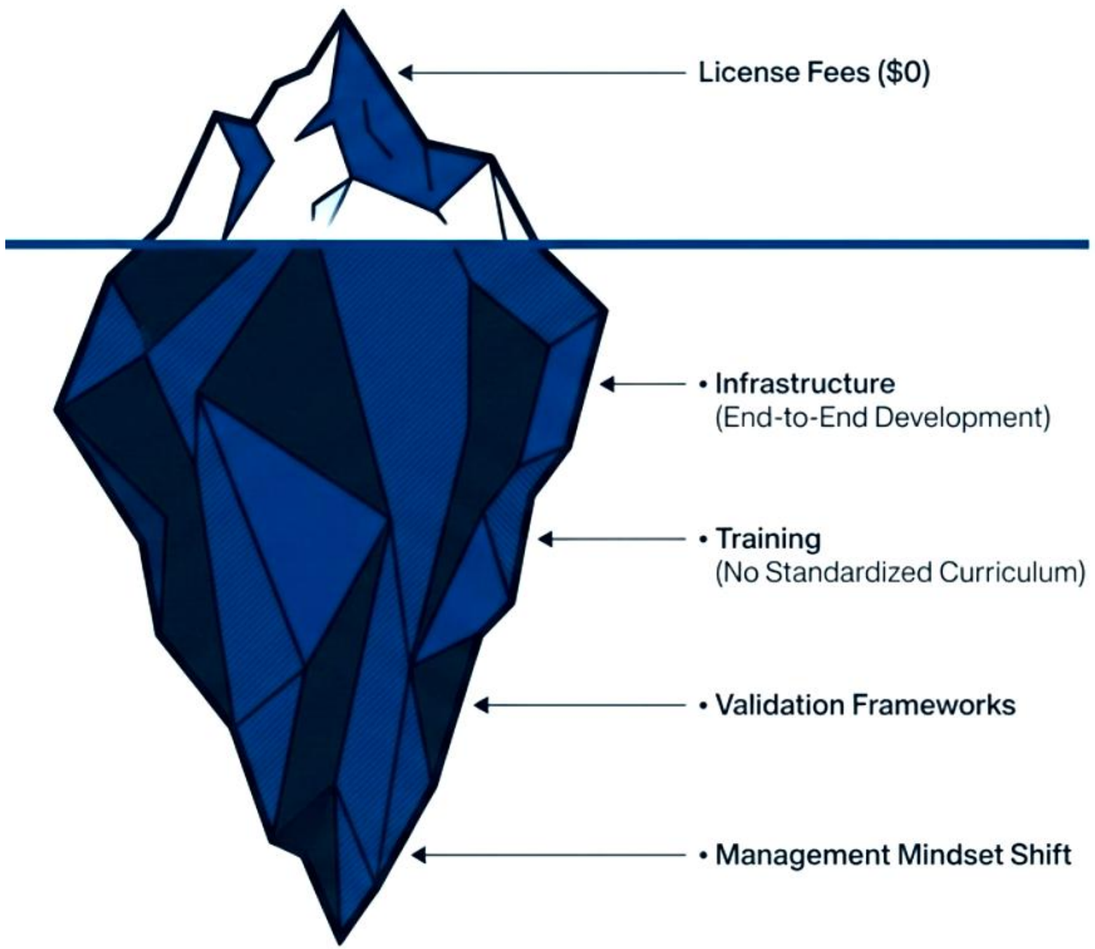
Barrier-breaking speed of the Concorde

Not just a software update. A different machine entirely.

Challenges to Adoption: A Bumpy Ride



Challenges to Adoption: No Plug-n-Play



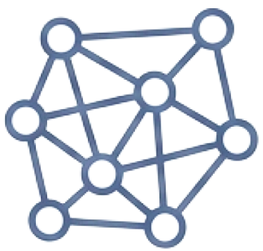
The Hidden Costs of "Free"

Unlike SAS, there is no standardized training. Organizations must build their own curriculums while managers adapt to new oversight models.

What We Need: A Structured Assessment

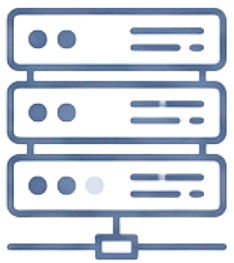
The Readiness Framework: Four Dimensions of Control

Success is an integrated ecosystem. You cannot have one without the others.



PEOPLE

- Skills & Training
- Cross-Functional Communities
- Leadership Sponsorship

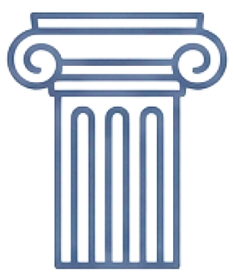


INFRASTRUCTURE

- Blessed Environments
- Containerization
- Security & Scalability

PROCESS

- Risk-Based Validation (CSA)
- Version Control (Git)
- Workflows



GOVERNANCE

- Policies & Roles
- Risk Management
- Licensing Strategy

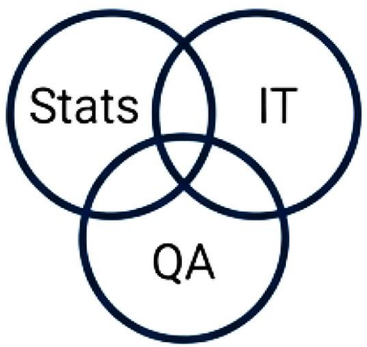
Dimension 1: People & Culture

Technology fails without the human element.

Compliance Training

Data scientists need training not just in coding, but in compliance expectations. Understanding validation and documentation is mandatory.

Cross-Functional Communities



Break down silos. Success requires explicit alignment between Statisticians, Programmers, and Quality Assurance.

Executive Sponsorship

Strong leadership is required to drive change management, secure resources, and champion the shift to Open Source.

1 Dimension 1: People – Cultivating the Talent, Leadership, and Culture for Success

Skills & Training

Develop clear plans for upskilling statisticians, programmers, and QA. Define required competencies beyond just coding.

Leadership & Sponsorship

Secure executive-level champions who own accountability for the open-source strategy and can resolve cross-functional conflicts.

Culture of Collaboration

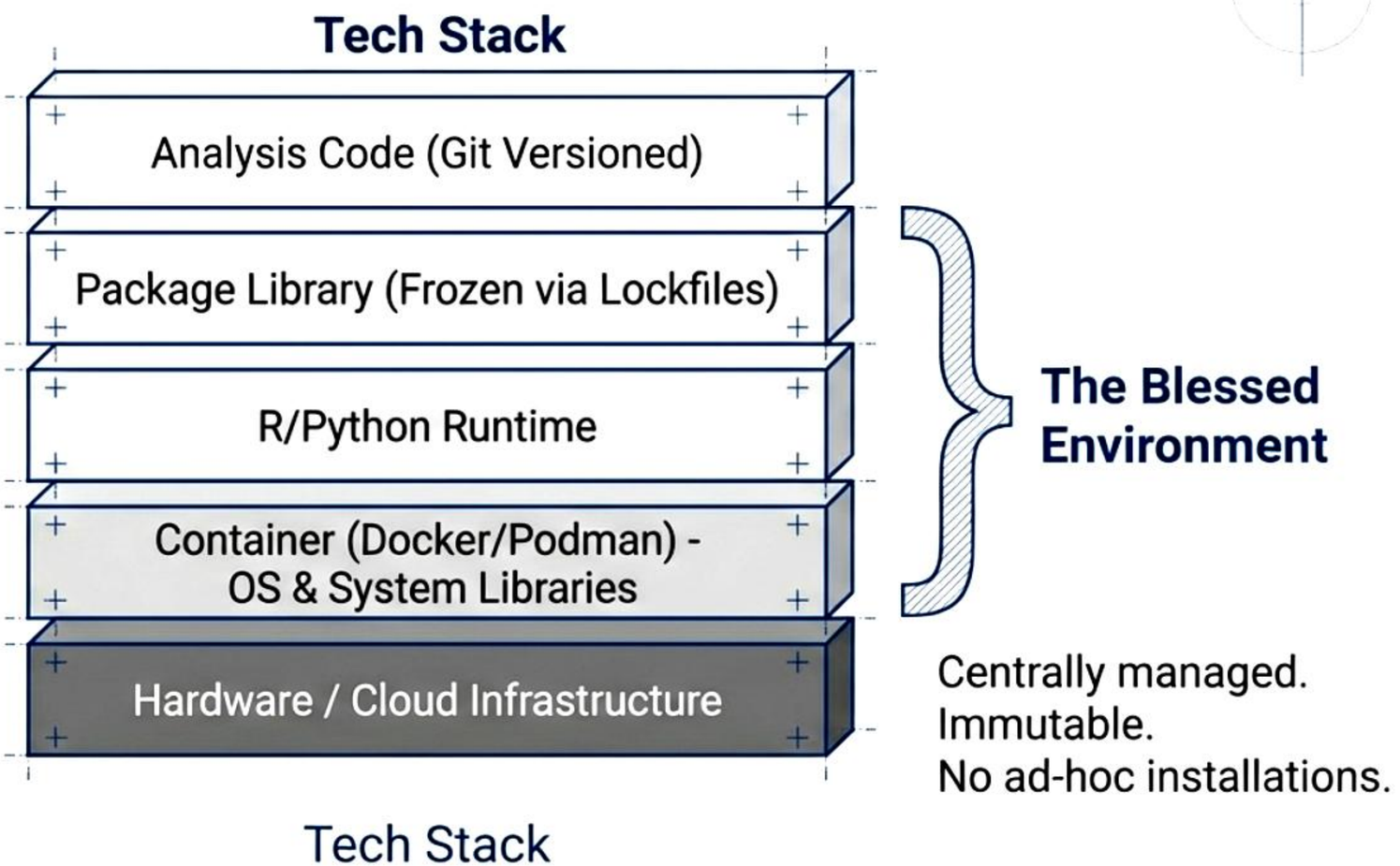
Foster internal communities of practice and cross-functional teams (e.g., statisticians, programmers, IT, QA) to share knowledge and build internal champions, as demonstrated by Roche.

Key Readiness Questions

- Does our leadership visibly champion and fund open-source use?
- How are we training teams and splitting roles between R and SAS, like Johnson & Johnson's hybrid model?
- Do we have a formal structure for internal communities

Dimension 2: Infrastructure & Reproducibility

The Golden Rule: **If you can't reproduce** it 5 years later, it isn't compliant.



2 Dimension 2: Infrastructure – Building a Secure, Reproducible, and Scalable Environment

Computing Environment

Standardize environments using technologies like containers (e.g., Docker, Podman) to ensure analyses are portable and auditable.

Security & Access Controls

Implement role-based access, audit trails, and controls for PHI/PII to meet HIPAA/GDPR and 21 CFR Part 11 expectations.

Reproducibility

Enforce version-controlled, 'blessed' environments with frozen, documented package snapshots (e.g., using `renv` lockfiles) for each submission.

Key Readiness Questions

- Can a regulator precisely reproduce our results using the environment we provide?
- How do we manage and document package dependencies for a specific submission?
- Are our audit trails sufficient to reconstruct an analysis from code to result?

PHI – Protected Health Information | PII – Personally Identifiable Information

Dimension 3: Process & Risk-Based Validation

Shift from Computer System Validation (CSV) to **Computer Software Assurance (CSA)**

Primary Endpoints / Safety Data

- **Action:** Rigorous, scripted testing (PQ/UAT).

Exploratory / Visualization

- **Action:** Streamlined documentation. Rely on community testing.

Package Validation Strategy

Leverage community risk metrics (e.g., R Validation Hub) to score packages on Maintenance, Usage, and Security. Validate the **ENVIRONMENT**, not just the package.

PQ – Performance Qualification | UAT – User Acceptance Testing

3 Dimension 3: Process – Defining Validated Workflows from Code to Submission

Risk-Based Validation

Adopt Computer Software Assurance (CSA) principles. Scale validation effort to the risk of the tool's intended use (e.g., more rigorous testing for primary endpoint analysis).

Version Control Integration

Mandate Git-based version control for all analytical code, institutionalizing practices like peer review and traceable deployment pipelines.

Clinical Workflow Integration

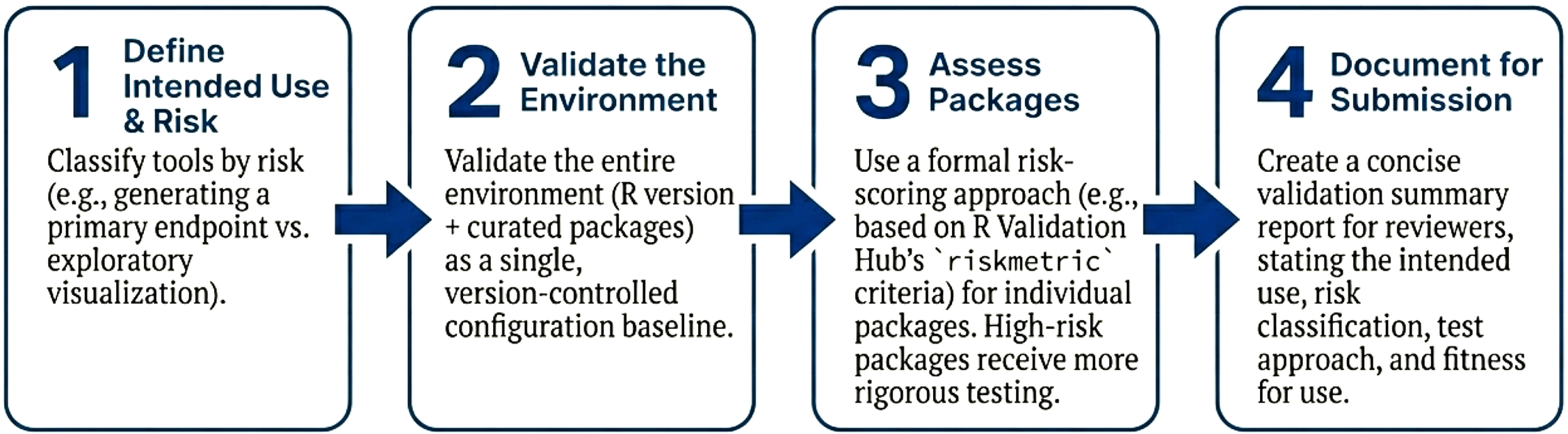
Ensure open-source tools fit within existing SDTM/ADaM workflows and can produce CDISC-compliant datasets and submission-ready documentation.

Key Readiness Questions

- Is our package validation approach documented, risk-based, and aligned with GxP principles?
- Are our coding practices integrated with version control from the start?
- How do we demonstrate that our R-based outputs are CDISC-compliant?

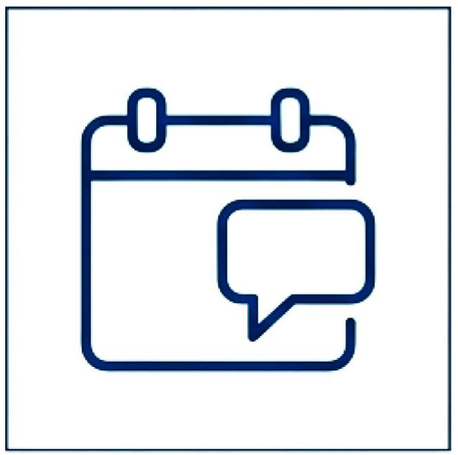
Answering the Regulator: A Risk-Based Approach to Validation

Validation focuses on ensuring “fitness for intended use” through a risk-based approach, aligned with FDA’s Computer Software Assurance (CSA) guidance.



A Proactive Strategy for Regulatory Engagement

Best practices based on successful filings:



Engage Early & Explicitly

Discuss the planned use of R and Shiny in pre-submission meetings (e.g., Type C meetings). Don't let it be a surprise.

Document Clearly in the ADRG

Detail the R version, key packages (including internal ones), and the environment snapshot in the Analysis Data Reviewer's Guide.

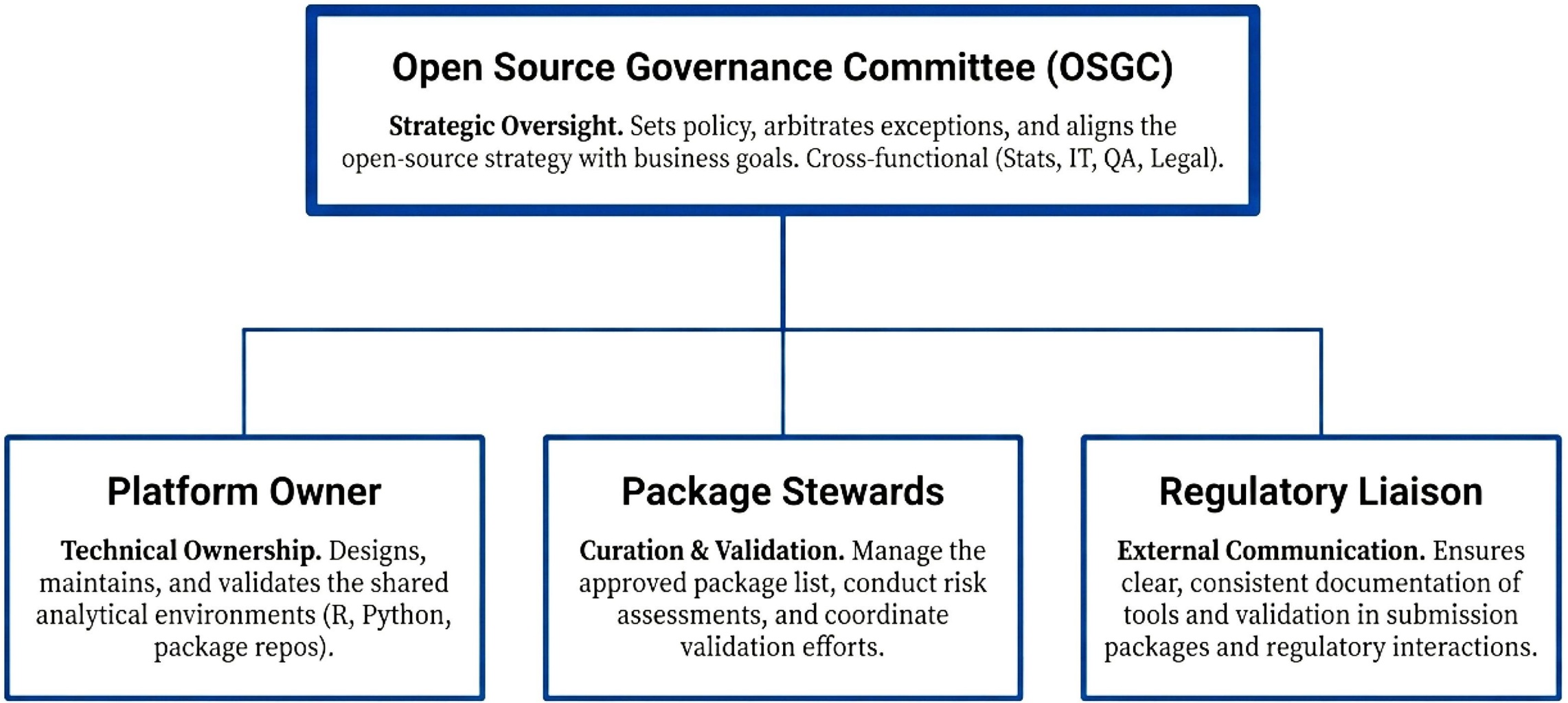
Ensure Full Reproducibility

Provide reviewers with everything needed to re-run analyses: well-documented code, environment lockfiles (`renv.lock`), and clear setup instructions.

Be Transparent About Validation

Document the validation strategy for all tools, providing a summary for external packages and a more detailed report for custom-developed internal packages.

Dimension 4: **Governance** & Accountability



4 Dimension 4: Governance – Establishing Clear Rules, Roles, and Risk Management

Policies & Strategy

- Establish an enterprise policy defining where open-source is allowed (e.g., exploratory vs. regulatory submissions) and classifying use cases by risk.

Roles & Responsibilities

- Create a cross-functional Open Source Governance Committee (OSGC) with clear ownership for package curation, environment maintenance, and validation decisions.

License & Risk Management

- Define acceptable software licenses to manage legal and IP risk. Maintain a risk register for key open-source components.

?

Key Readiness Questions

- 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?

Managing Legal and IP Risk: A Practical Approach to Open-Source Licensing

A formal policy on acceptable licenses is crucial to avoid IP contamination, ensure long-term maintenance rights, and manage compliance.

Preferred / Permissive

Minimal restrictions, easy to combine with proprietary code.

MIT, Apache 2.0, BSD

Allowed with Review

Weak copyleft licenses; require legal/architectural review, especially if code is redistributed.

LGPL, MPL, EPL

Highly Restricted / Exception-Only

Strong "viral" copyleft licenses; pose significant risk to proprietary code and require explicit legal approval and containment strategies.

GPL, AGPL

GPL – GNU General Public License | LGPL – Lesser GPL | MPL – Mozilla Public License | EPL – Eclipse Public License | AGPL – Affero GPL

Your 90-Day Roadmap: From Fear to Strategy

The question is no longer IF we can use open source, but HOW we prepare to use it systematically.

Measuring Success: Program Health Metrics

Adoption & Value

% of clinical projects using approved R workflows vs. legacy tools.

Compliance & Risk

# of open, high/critical CVEs in approved components.

Vulnerability Remediation

Time to remediate vulnerabilities in validated environments.

Operational Health

Average cycle time for new package approval (from request to availability).

Regulatory Success

# of open source-related findings in regulatory audits or inspections.

CVEs - Common Vulnerabilities and Exposures

Recommendation: Avoid The Validation Fallacy

Source: MIT Media Lab Experiment

FINDING 1:

Many peers do not review code closely.

FINDING 2:

Most are not fully qualified to do so.

THE IMPLICATION:

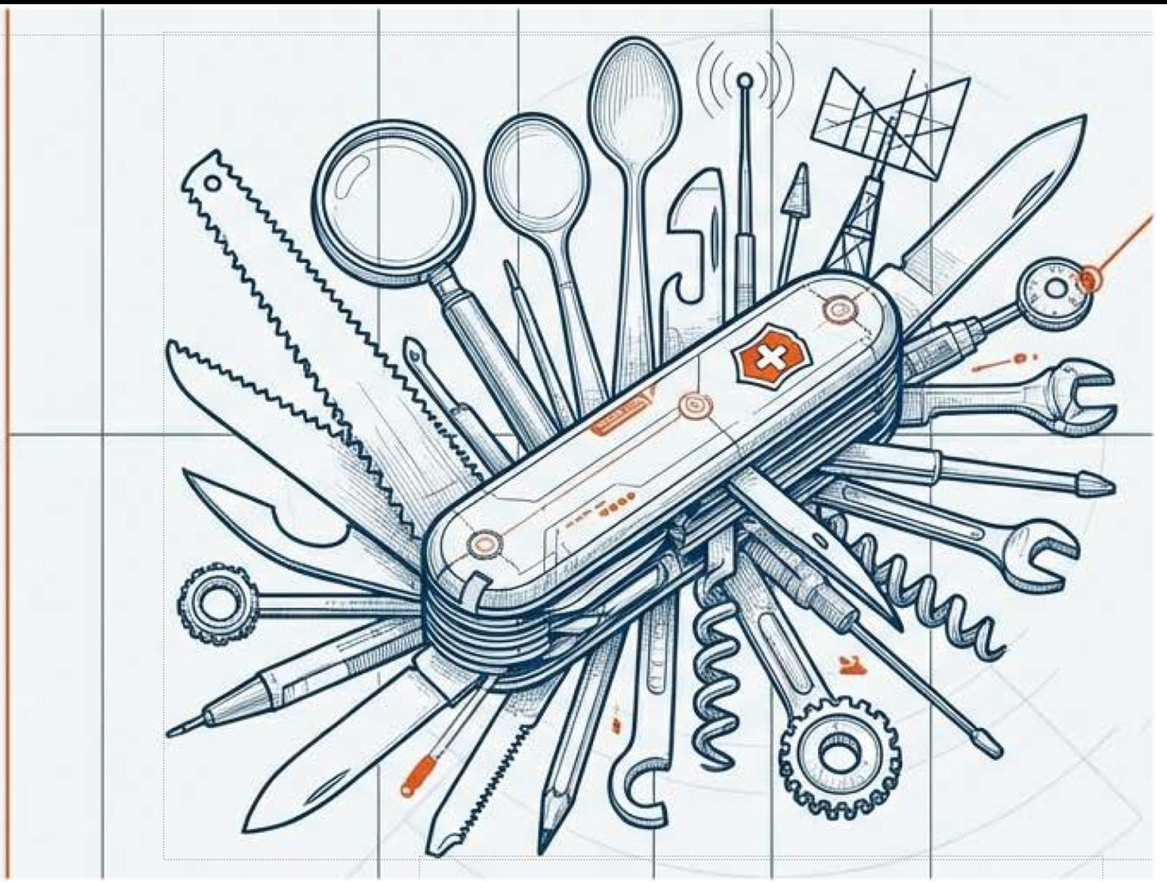
Do not assume “Open Source” means “Vetted”.

Recommendation: Choose Open-Source Solutions Wisely - Avoid The Swiss Army Knife Trap

FROM BASIC VERSION CONTROL TO GRANULAR TRACKING

FROM MANAGERIAL OVERSIGHT TO DISTRIBUTED CODE REVIEW

FROM STATIC MACROS TO CONTINUOUS INTEGRATION



“Engineered to do everything. Optimized for nothing.”

BAND-AIDS

Relying on bespoke fixes to force compatibility with legacy systems.

TIGHT INTEGRATION

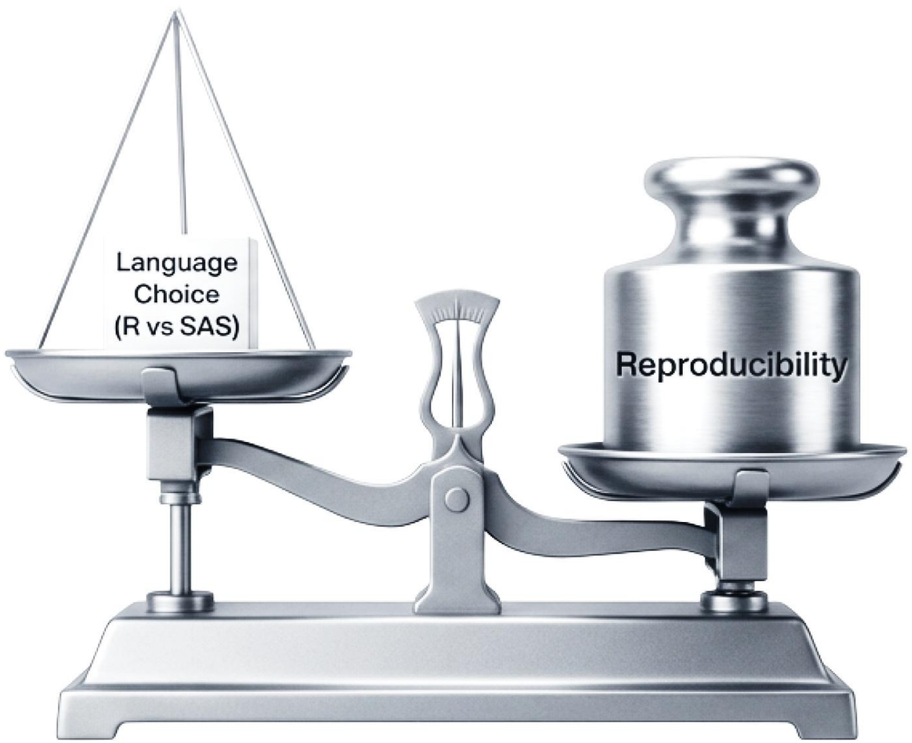
Fundamental deviation from the portable philosophy of open-source.

THE RESULT

Technical debt that is difficult to replicate and rely on.

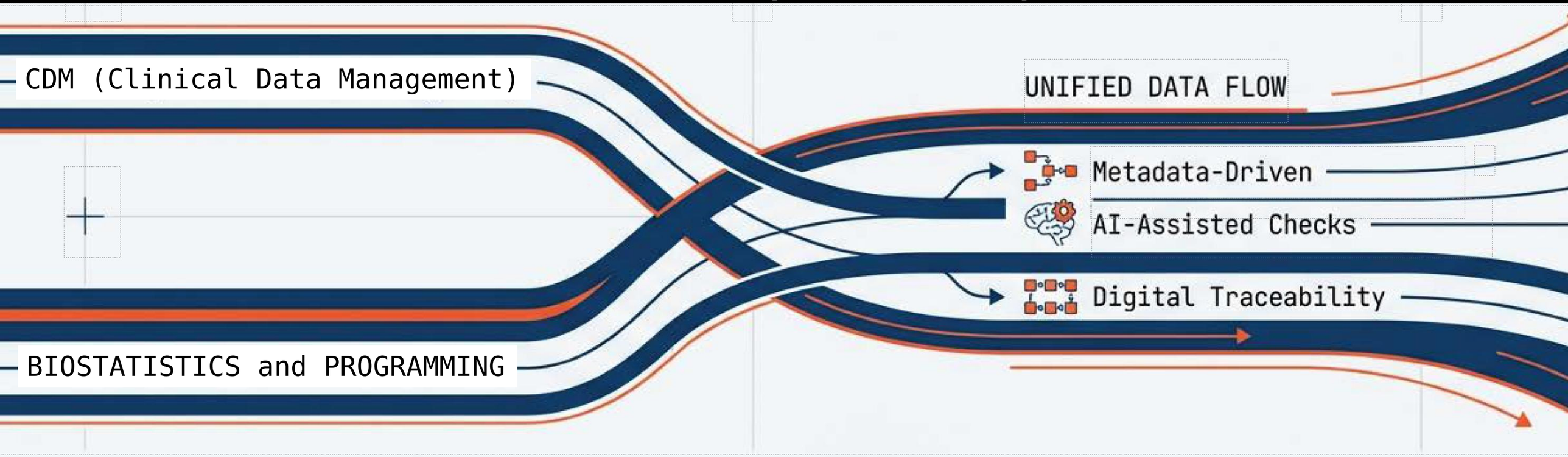
Recommendation: Reproducibility is Key for Regulatory

The Big Secret: Regulators Care About Reproducibility, Not Syntax



- Reality: FDA feedback proves they do not judge based on tool choice, but on the ability to re-run analysis years later.
- Evidence: In the Novo Nordisk submission, FDA requested code with minimal dependencies to simplify review.
- Requirement: Success requires frozen, documented snapshots of the environment (containers/lockfiles).

Looking Ahead: Enabling Data Without Boundaries with End-to-End Open-Source Adoption



FROM:

Handoffs and Silos

- Manual Data Transfers
- Disconnected Systems
- Reactive Issue Resolution

TO:

Shared Accountability & Automation

- Real-Time Collaboration
- Integrated Workflows
- Proactive Quality Control

RESULT:

Accelerated Database Lock to Submission

- Faster Time-to-Insight
- Reduced Cycle Times
- Enhanced Data Integrity

Looking Ahead: People, People, People!

If you want to go fast, go alone. If you want to go far, go together.

R We Ready? A Framework for Open-Source Adoption in Pharma

Build a Skilled & Supported Team

Focus on training, defining roles, and securing leadership buy-in for a collaborative culture.

FOUNDATIONAL PILLARS:
PEOPLE & INFRASTRUCTURE

Establish a Secure & Reproducible Environment

Use controlled, versioned computing environments (e.g., containers) with robust security and audit trails.

Standardize Workflows & Validation

Implement GxP-aligned, risk-based validation for packages and integrate with version control.

OPERATIONAL FRAMEWORK:
PROCESS & GOVERNANCE

Define Clear Ownership & Policies

Create a governance committee to manage policies, roles, risk, and compliance.

Ecosystem Resources & References

Key Industry Initiatives

- **Pharmaverse**: Curated list of open-source R packages for clinical reporting.
- **R Validation Hub**: Tools for assessing package risk (riskmetric).
- **PHUSE**: Working groups on ‘Data Visualization’ and ‘Open Source guidance’.
- **R Consortium**: Sponsors of the FDA Submission Pilots.

Recommended Reading

- “A Pharma Guide to Safely Adopting Open-Source Software”
- FDA Pilot 4 Findings on WebAssembly
- **GSK** / **Appsilon** White Papers on GxP Compliance