PharmaSUG 2026 - Paper ET-370

Challenges for Small- to Mid-Size Organizations to Build a GxP-Compliant R Environment (CRE)

Peilin Zhou; Peng Zhang, PhD; Tai Xie, PhD; Christine Matakovich, CIMS Global, Somerset, NJ

ABSTRACT

Small- to mid-size organizations are increasingly adopting open-source technologies such as R for clinical trial analysis due to their flexibility and cost efficiency. However, building a GxP-compliant R computing environment (CRE) presents significant challenges, particularly in areas such as package governance, validation, reproducibility, and resource constraints. These challenges are further amplified when integrating emerging technologies such as large language models (LLMs), which introduce additional concerns regarding statistical validity, traceability, and controlled execution.

This paper discusses key challenges faced by small- to mid-size organizations in establishing a compliant and scalable CRE, including management of external and internal R packages, implementation of validation frameworks, and ensuring statistical consistency across programming environments. To address these challenges, a structured approach is proposed that combines a controlled CRE, validated package management workflows, formal validation practices, and a constrained LLM-driven analysis workflow.

Within this framework, LLMs are integrated as an interface layer that maps user queries to predefined, validated statistical functions, ensuring reproducibility and auditability. By aligning technical solutions with regulatory expectations, this approach provides a practical pathway for organizations with limited resources to adopt open-source and AI-driven technologies while maintaining statistical integrity and compliance.

INTRODUCTION

Open-source tools such as R are increasingly being adopted in clinical trial analysis, reporting, and regulatory submission due to their flexibility, extensibility, and strong community support. For small- to mid-size organizations, open-source solutions offer a cost-effective alternative to proprietary systems while enabling innovation in areas such as automation, interactive reporting, and integration with emerging technologies.

However, building a GxP-compliant R computing environment (CRE) within these organizations presents several practical challenges. Limited resources, lack of standardized infrastructure, and the need to balance flexibility with regulatory compliance make it difficult to implement robust governance, validation, and reproducibility controls. In addition, managing external dependencies, developing internal tools, and ensuring statistical consistency across different programming environments require careful planning and technical expertise.

The emergence of large language models (LLMs) introduces further complexity. While LLMs can improve accessibility by enabling natural language interaction with data, they also raise concerns related to uncontrolled code execution, lack of transparency, and potential inconsistencies in statistical outputs.

This paper discusses the key challenges faced by small- to mid-size organizations in building a compliant CRE and proposes a structured framework to address them. The framework integrates controlled package management, formal validation practices, and a constrained LLM-driven workflow, enabling organizations to leverage modern technologies while maintaining statistical validity, reproducibility, and regulatory compliance.

KEY CHALLENGES FOR SMALL- TO MID-SIZE ORGANIZATIONS

Small- to mid-size organizations face several unique challenges when attempting to establish a GxP-compliant R computing environment (CRE), including resource limitations, gaps in open-source expertise, and uncertainty in applying GxP principles within an R-based ecosystem.

RESOURCE CONSTRAINTS

Unlike large organizations, smaller teams often lack dedicated infrastructure, validation teams, and governance frameworks. This makes it difficult to implement and maintain a fully validated computing environment.

LIMITED OPEN-SOURCE EXPERTISE

Small- to mid-size organizations often have limited experience working with open-source ecosystems such as R. Unlike proprietary software with standardized workflows and vendor support, open-source tools require users to evaluate package quality, understand community-driven development practices, and manage dependencies independently. This lack of familiarity can lead to inconsistent package selection, improper use of libraries, and increased risk in regulated environments.

PACKAGE GOVERNANCE COMPLEXITY

The open-source nature of R introduces challenges in identifying reliable packages, managing dependencies, and ensuring consistent usage across projects. Without proper controls, this can lead to reproducibility issues.

VALIDATION BURDEN

Establishing validation processes for packages, workflows, and applications requires significant effort. Smaller organizations may struggle to balance validation rigor with available resources.

CROSS-SYSTEM CONSISTENCY

Ensuring consistency between R and proprietary systems (for example, SAS) is critical for regulatory acceptance but can be technically challenging due to differences in implementation and defaults.

GXP INTERPRETATION IN R CONTEXT

While GxP principles are well established in traditional statistical computing environments, their interpretation in the context of open-source R is less standardized. Organizations may face challenges in translating regulatory expectations, such as validation, traceability, and auditability, into practical implementation within R-based workflows. This includes uncertainty in defining validation scope for packages, documenting evidence, and establishing compliant development and deployment practices.

INTEGRATION OF EMERGING TECHNOLOGIES

Incorporating LLMs introduces additional risks, including uncontrolled execution, lack of traceability, and potential misinterpretation of statistical logic.

These challenges highlight the need for a structured yet practical approach that balances compliance requirements with operational feasibility.

The following sections describe a structured framework designed to address these challenges through controlled environments, validated workflows, and constrained integration of emerging technologies.

GXP COMPLIANT R ENVIRONMENT (CRE)

To address infrastructure and governance challenges, a GxP-compliant R Environment (CRE) serves as the foundational component.

A Compliant R Environment (CRE) provides the foundation for adopting open-source tools in regulated clinical workflows. As outlined in the PHUSE Statistical Computing Environment white paper [1], a CRE must enforce governance, reproducibility, and traceability.

Key components include:

- Role-Based Access Control (RBAC): Fine-grained permissions at organization, project, and user levels
- Audit Trails: Full tracking of user activities and system-level changes to support 21 CFR Part 11 compliance
- Validated Infrastructure: Controlled environments ensuring reproducibility across analyses
- Curated Package Repository: Approved packages with documented risk assessment and validation
- Secure Execution Environment: Typically Linux-based systems for enhanced security and stability

With these controls, the CRE enables consistent R versioning, standardized workflows, and secure multi-project execution. It also supports Shiny deployment and collaborative review processes, which are essential for modern clinical data analysis.

In practice, the CRE ensures that all users operate within a standardized analytical environment, where package versions, system configurations, and access permissions are centrally controlled. This minimizes variability across studies and users, enabling consistent and reproducible results throughout the organization. By enforcing these controls at the infrastructure level, the CRE serves as the foundation upon which validation, package management, and LLM-based workflows can reliably operate.

EXTERNAL R PACKAGE MANAGEMENT

To address challenges related to package governance and reproducibility, a controlled external package management workflow is required.

Open-source R provides access to a vast ecosystem of packages (for example, pharmaverse [2]), but their use in regulated environments requires strict governance.

Key challenges include:

- Identifying reliable and appropriate packages
- Managing version control and dependencies
- Ensuring reproducibility across environments

To address these challenges, organizations should implement a validated package qualification workflow, including:

- Risk Assessment: Tools such as {riskmetric} can quantify package quality based on documentation, testing, and maintenance
- Dependency Control: Implementation of controlled mechanisms to manage and lock both R package dependencies and underlying system dependencies, ensuring reproducibility, environment consistency, and stability.
- Curated Repository: Internal repositories containing only approved packages

Only qualified packages are promoted into the CRE, ensuring that all analyses rely on validated and controlled tools. Figure 1 illustrates the end-to-end package management workflow. See the R Package Management White Paper [3] for further guidance.

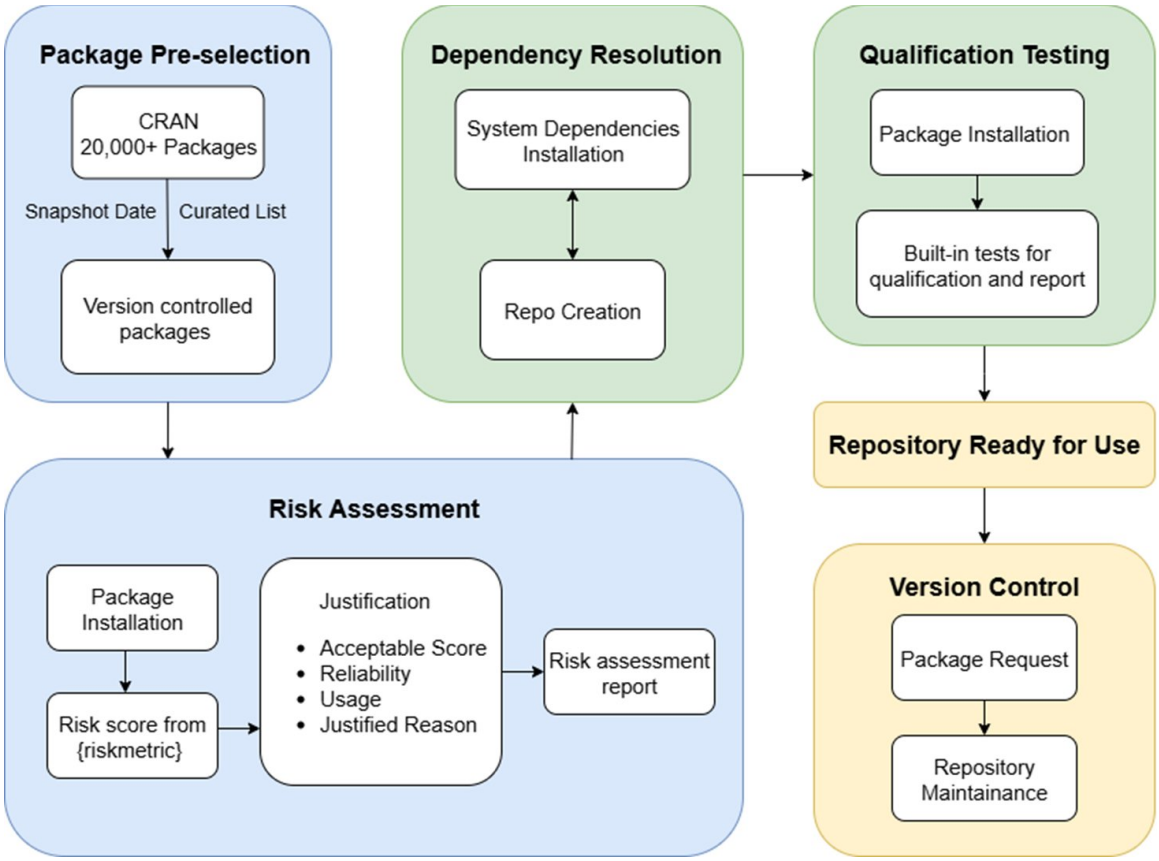

Figure 1. External R Package Management Workflow

Figure 1 illustrates the end-to-end workflow for external R package management within a GxP-compliant R computing environment (CRE). The workflow begins with the identification of candidate packages from open-source repositories (for example, CRAN), followed by a structured evaluation process.

Each candidate package undergoes a risk assessment to evaluate factors such as documentation quality, test coverage, maintenance activity, and community adoption. Based on this assessment, packages are classified and either approved, conditionally approved with justification, or rejected. Approved packages proceed to a controlled environment where their versions and dependencies are locked to ensure reproducibility.

Once qualified, packages are incorporated into an internal curated repository (for example, miniCRAN), which serves as the single source of truth for all analytical activities within the organization. This repository ensures that all users operate with consistent and validated package versions across projects.

The workflow also includes governance and maintenance steps, such as periodic re-evaluation of packages, monitoring for updates or deprecations, and documentation of any changes. These steps ensure that the package ecosystem remains stable, compliant, and aligned with regulatory expectations over time.

By enforcing both technical controls and governance processes, this workflow enables organizations to leverage open-source R packages while maintaining reproducibility, traceability, and regulatory compliance.

INTERNAL PACKAGE DEVELOPMENT

In addition to external packages, organizations often develop internal R packages to standardize workflows for ADaM dataset derivation, TLF generation, and study-specific statistical analyses.

A standardized development lifecycle is essential to ensure quality and compliance. Based on best practices from {openstatsguide} [4], this includes:

- Requirement definition aligned with the SAP
- Environment control using {renv}
- Documentation with {roxygen2}
- Version control via Git
- Unit testing using {testthat}
- CI/CD pipelines for automated validation
- Formal validation reports (for example, via {valtools})

These practices ensure that internal tools are reusable, scalable, and compliant across studies.

VALIDATION AND STATISTICAL VALIDITY

To mitigate validation burden and ensure statistical correctness, structured validation strategies must be implemented.

Ensuring statistical validity is central to adopting open-source tools in clinical trials. Three key areas must be addressed.

VALIDATION PER REQUIREMENT

Each package or function must be validated against predefined requirements. Validation frameworks such as {valtools} [5] support test case definition, execution and documentation, and generation of validation reports. These reports serve as evidence during regulatory audits.

EVIDENCE-BASED TESTING FOR SHINY APPLICATIONS

Shiny applications enable interactive data exploration but must be rigorously tested to ensure correct mapping between user inputs and statistical functions, and accurate rendering of outputs. Testing should confirm that results are identical to those generated by validated backend functions. All validation evidence must be retained in controlled repositories.

STATISTICAL VALIDITY ACROSS PROGRAMMING LANGUAGES

Differences between R and proprietary software (for example, SAS) may arise due to algorithm implementation differences, numerical precision, or default parameter settings. For example, survival analysis methods may differ in tie handling (Efron vs. Breslow). Therefore, results must be cross-validated, methodologies must align with the SAP, and any discrepancies must be documented and justified.

ANALYSIS RESULT FROM LLM

To safely incorporate emerging technologies while maintaining control, a constrained LLM-driven workflow is proposed.

LLMs introduce a powerful interface for querying clinical data but must be integrated carefully to avoid risks such as uncontrolled code execution and invalid statistical outputs.

WORKFLOW ARCHITECTURE

The proposed workflow introduces a controlled execution layer between the LLM and the data:

- Natural Language Interpretation: The LLM interprets user queries and identifies relevant data concepts

- Controlled Code Generation: The LLM maps user intent to predefined, validated functions rather than generating arbitrary code
- Execution via Validated Functions: Only approved functions from the curated package library are executed
- Output Rendering: Results are converted into human-readable outputs using template-based reporting

This design ensures that the LLM acts as an orchestration layer rather than a computation engine, preserving control over statistical methodology while improving accessibility for end users.

RETRIEVAL-AUGMENTED GENERATION (RAG)

To improve domain-specific accuracy, RAG is used to provide contextual knowledge from Statistical Analysis Plans (SAP), data dictionaries, and standard analysis templates. R packages such as {ellmer} and {ragnar} can support this integration. This ensures that generated outputs are aligned with study-specific requirements.

RISK MITIGATION AND GOVERNANCE

To ensure compliance, the following controls are enforced:

- No direct execution of arbitrary LLM-generated code
- Restricted access to validated function libraries only
- Full traceability of executed code and outputs
- Audit logging of all interactions
- Human review for critical outputs

It is important to note that LLM-generated outputs are not intended to replace validated statistical programming workflows. Instead, they serve as an interface layer to facilitate data exploration and communication. All critical analyses must continue to rely on validated programs and undergo standard review and quality control procedures.

CONCLUSION

The integration of open-source R and large language models presents significant opportunities to enhance efficiency, accessibility, and innovation in clinical trial analysis. However, these technologies must be adopted within a structured and validated framework to meet regulatory expectations.

This paper demonstrates that a combination of GxP-compliant computing environments, controlled package management and development practices, rigorous validation strategies, and a constrained LLM execution architecture can enable safe and scalable adoption of these technologies.

By addressing the specific challenges faced by small- to mid-size organizations, this framework provides a practical pathway to leverage modern analytical tools while maintaining statistical validity, reproducibility, and compliance.

REFERENCES

[1] PHUSE White Paper, Statistical Computing Environment. Available at: https://www.lexjansen.com/phuse-us/2021/hw/PAP_HoW04.pdf

[2] {pharmaverse}. Available at: https://pharmaverse.org/

[3] R Package Management White Paper. Available at: https://pharmar.org/white-paper/

[4] {openstatsguide}. Available at: https://www.openstatsware.org/guide.html

[5] R Package Validation Framework. Available at: https://phuse.s3.eu-central-1.amazonaws.com/Deliverables/Data+Visualisation+%26+Open+Source+Technology/WP059.pdf

CONTACT INFORMATION

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

Peilin Zhou
CIMS Global, Somerset, NJ
909-680-2068
pzhou@cims-global.com
www.cims-global.com