PharmaSUG China 2026 - Paper DS-224

R Submissions in Practice: The J&J Implementation and Emerging AI Opportunities

Wenlan Pan, Johnson & Johnson

ABSTRACT

The question for R-based regulatory submissions is no longer whether they work, but how to make them work reliably, repeatedly, and at scale. With successful filings now reported across multiple sponsors and health authorities, the field has the evidence needed to move from feasibility to execution, focusing on implementation decisions, documentation standards, and where generative AI may responsibly reduce effort. This paper addresses those questions in two connected parts. First, it documents the Johnson & Johnson submission experience in depth, covering package management and governance, ADRG design, eCTD file structure adaptations, mock submission practice, and adaptations for multi-regional filings including China's CDE, with key enablers described as reusable design patterns applicable across sponsors and therapeutic areas. Second, it explores where generative AI may fit within the submission programming workflow to reduce manual effort in repetitive and documentation-heavy tasks, and discusses the controls needed to keep AI-assisted work compliant and reviewable, including traceability, reproducibility, and human oversight requirements. By combining an in-depth implementation case study with an exploratory AI assessment, this paper provides a practical reference for teams building scalable R-based workflows and a forward-looking perspective on how AI may enhance efficiency without compromising regulatory compliance. It is designed to be read alongside the companion paper on health authority landscape analysis and AI application frameworks.

INTRODUCTION

R has moved from exploratory use to a production option for regulatory submissions. R Consortium pilots have shown that R-generated tables, listings, and figures (TLFs), ADaM data sets, and newer data-delivery formats can pass through established submission channels (R Consortium 2023, 2026). FDA's current eCTD file-format specification accepts R scripts and the additional file types used to deliver R packages (FDA 2025a). Industry adoption has advanced in parallel, with sponsors have reported both hybrid SAS/R and end-to-end R filings across multiple health authorities. The practical question is therefore no longer whether R can be submitted, but how to make R deliverables reliable, reproducible, and easy to review.

This paper examines that question through the Johnson & Johnson (J&J) implementation, drawing on two studies from different therapeutic areas that used a hybrid model: SAS generated the submitted ADaM data sets and R generated all the submitted TLFs. Part I examines the operating controls behind this production-scale use: package governance, recognizable program design, cross-language verification, an R-specific Analysis Data Reviewer's Guide (ADRG), submission packaging, mock-submission practice, and regional adaptation. These are presented as reusable design patterns rather than a single prescribed architecture.

Part II turns to the same lifecycle to ask where generative AI can responsibly reduce effort. It holds the acceptance model fixed, deterministic checks and qualified human sign-off, and treats AI as a source of reviewable candidates rather than finished deliverables. The connection between the two parts is direct: the structured evidence that disciplined submission work already produces is exactly what an AI assistant needs, both as grounding for its drafts and as the basis for checking them. The companion paper (Liu 2026) covers the broader health-authority expectations and AI frameworks; this paper stays implementation-focused.

PART I. R SUBMISSIONS IN PRACTICE: THE J&J IMPLEMENTATION

Successful R-based submissions depend less on the programming language than on implementation design. Four requirements shape that design: the workflow must generate correct results, control the software environment, retain independent verification evidence, and give the reviewers enough

information to reproduce the outputs outside the sponsor's systems. Figure 1 shows how the J&J workflow met these requirements end to end, from a validated R production environment through to regulatory submission.

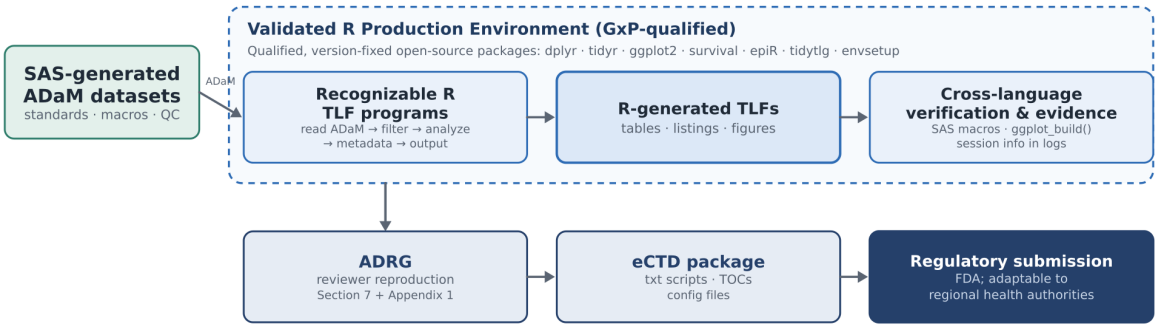

Figure 1. The Hybrid SAS/R Submission Workflow and Its Control Environment

A PRAGMATIC HYBRID SAS/R STRATEGY

The first design decision was where R could add value without rebuilding the entire analysis pipeline. J&J kept submitted ADaM production in SAS, preserving established standards, in-house macros, mature QC, and institutional knowledge, and moved all submitted TLFs to R for its flexible reporting, advanced visualization, and active open-source ecosystem.

Derivations were retained in ADaM whenever practical, so each TLF could be generated directly from standards-compliant analysis data rather than through extensive downstream transformation in R. This kept the R reporting layer thin, reduced study-specific helper codes, and strengthened traceability from every output back to its supporting analysis data.

Assigning each language to the layer where it was already strong let the team introduce R at production scale where it provided clear value while holding operational risk low.

CONTROLLED PACKAGE ENVIRONMENT AND PROGRAM DESIGN

Package governance was treated as part of submission design. Where a mature SAS workflow relies on internally developed and maintained components, an R workflow draws much of its reusable functionality from open-source packages, so the relevant question was not merely whether the code worked but whether the package ecosystem could be made inspectable, controlled, and reproducible for review. In the J&J implementation, every R package used to generate analysis outputs was open source, so reviewers could inspect its source, but public availability did not substitute for sponsor control. Selected package versions were qualified through J&J's risk-based GxP process, fixed for production use, and delivered through controlled, containerized releases in the validated environment, combining transparency with a stable, documented package surface.

The production set combined widely used packages such as `dplyr`, `tidyr`, `ggplot2`, `survival`, and `epiR` with J&J-developed packages contributed to the `pharmaverse`: `envsetup` externalized input and output paths through configuration files, so that no study paths were hard-coded, and `tidytlg` supplied reusable TLF-building functions.

Reviewability depends on the submitted programs as much as on the final outputs, so every R TLF program followed a recognizable sequence: read the required ADaM inputs, apply the relevant population and analysis filters, perform the planned summary or statistical analysis, apply column metadata, and generate the output with titles and footnotes kept separate from the analysis logic. A representative script makes the pattern concrete:

```
library(dplyr); library(haven); library(envsetup); library(tidytlg)

# Read ADaM and apply the analysis population filter
```

```
adsl <- read_sas(read_path(a_in, "adsl.sas7bdat")) %>%
  filter(RANDFL == "Y", TRT01PN %in% c(1, 2))

# Build one section with a tidytlg descriptive function
t_age <- adsl %>%
  univar(colvar = "TRT01PN", rowvar = "AGE",
         statlist = statlist(c("N", "MEANSD", "MEDIAN", "RANGE")))

# Bind sections, apply column metadata, and write the output
tbl <- bind_table(t_age, t_sex, t_race,
         column_metadata_file = read_path(dpspath, "column_metadata.xlsx"))

gentlg(huxme = tbl, file = "tsidem01", orientation = "portrait",
       title_file = read_path(dpspath, "titles.xlsx"))
```

The goal was not identical scripts but recognizable ones: inputs, analysis logic, metadata, and output generation always sit in the same place, so peer review, reuse, QC, and reviewer navigation all become faster.

CROSS-LANGUAGE VERIFICATION AND REPRODUCIBILITY EVIDENCE

Verification was performed across languages, deliberately pairing established SAS standards with R. General summary tables were independently produced with standard SAS macros or programs, and automated utilities extracted and compared the contents of the SAS- and R-generated RTF files in batch. Figures required a different route: values were extracted from the underlying `ggplot2` object with functions such as `ggplot_build()` and written to the R log, where they could be checked against the corresponding tabular values or source data. For critical statistical methods, R results were verified against the equivalent SAS procedures, and the study team reviewed any discrepancy to decide whether it affected interpretation or conclusions.

Verification did more than surface discrepancies; it left durable reproducibility evidence. Each program's logic is documented, each result independently confirmed, and R session information captured in the log records, the exact package versions and environment behind every output. The lesson is not that two languages are permanently required, but that independent verification and durable evidence are required; cross-language QC was simply an efficient way to achieve both here.

ADRG DESIGN FOR REVIEWER REPRODUCTION

The ADRG is where a hybrid SAS/R submission earns a reviewer's confidence. A traditional SAS package can assume a familiar execution model; an R package cannot. The ADRG therefore becomes more than supporting documentation: it is the reviewer's practical guide to the submitted programs and to reproducing the R-generated outputs outside the sponsor's environment. Building on the R Consortium Pilot 3 template, the J&J ADRG expanded two areas for the hybrid submission (Figure 2): the R-specific

program inventory in Section 7, and Appendix 1 for R environment set-up and execution.

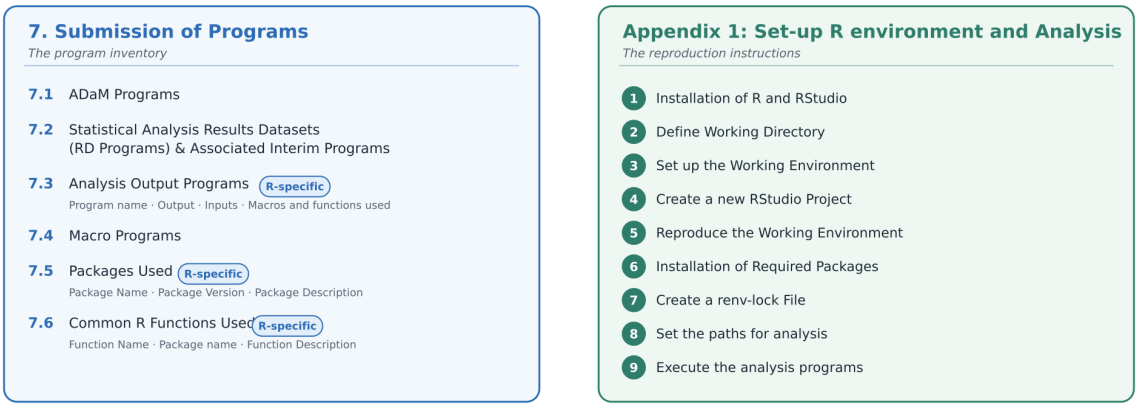

Figure 2. The R-specific Program Inventory (Section 7) and Reviewer-reproduction Workflow (Appendix 1) in the ADRG

The harder of the two additions is reviewer reproduction, and its central lesson is to treat the reviewer environment as a target in its own right rather than a copy of the sponsor's. Paths to data and helper functions are supplied separately through an `envsetup.txt` file added to the reviewer's `.Rprofile`, keeping environment-specific detail out of the analysis programs. Package installation runs from a submitted `packages.txt` file against a pinned repository snapshot, and the `renv.lock` file is generated locally on the reviewer's system after installation rather than shipped from the sponsor's Linux environment, preserving a reproducible, project-level package record while avoiding cross-platform portability problems. Absolute paths and exact package names are documented explicitly, reflecting Pilot 3 feedback and internal testing. The target is functional reproduction of the submitted outputs, not a byte-for-byte copy of the sponsor platform.

PACKAGING, AGENCY INTERACTION, AND REGIONAL ADAPTATION

For the two FDA filings, SAS and R source programs were submitted as text files with language-specific suffixes, for example `adsl-sas.txt` and `tsidem01-r.txt` and separate tables-of-contents distinguished SAS programs, SAS macros, R programs, and R functions, alongside the configuration files required by the ADRG appendix. Because accepted formats continue to evolve, file conversion and package assembly should be re-confirmed against the current file-format specification for each filing (FDA 2025a).

Early engagement with the agency was a further control. The two studies came from different therapeutic areas and review divisions. For Submission 1, a mock package gave an early check of file naming, dependencies, reviewer setup, and ADRG usability, and its feedback was incorporated before the real filing. Submission 2 followed the same hybrid strategy but, under a different division, did not include a mock eCTD. The value of a mock submission is practical: it surfaces missing dependencies, unclear instructions, validation issues, and reviewer-usability problems while they are still less costly to fix. Table 1 summarizes the interaction and submitted content for both studies.

Stage	Submission 1	Submission 2
Briefing book	ADaM data sets (xpt); SAS and R scripts (txt); ADRG with R package and version information; mock eCTD requested and accepted	ADaM data sets (xpt); SAS and R scripts (txt); ADRG with R package and version information
Mock eCTD	Dummy ADaM data sets (xpt) and five R table/figure scripts (txt); new J&J ADRG	Not part of the submission interaction

Stage	Submission 1	Submission 2
	template with R package/version information and R-environment recreation; validated and submitted through the FDA gateway	
Real submission	All ADaM data sets and MMRM results data sets (xpt); all ADaM SAS macros/programs (txt); 200+ R TLF scripts (txt); ADRG with study-specific R sections; validated eCTD	All ADaM data sets (xpt); all ADaM SAS macros/programs (txt); 300+ R TLF scripts (txt); ADRG with study-specific R sections; validated eCTD

Table 1. Scope and Interaction Milestones for the Two FDA Submissions

The implementation was also adapted for other regions. In the project-specific CDE communication described here, a hybrid SAS/R package was considered acceptable when organized under the clinical-trial data-submission guidance, while mock submission was not included in the standard acceptance process. The filing used a full Chinese ADRG, CDE-oriented folder mapping (for example, a `dataset` folder in place of `misc`), and XPT V5/V8 conversion support where needed, with labels and metadata localized as required. These are project-specific details rather than general CDE requirements, but the reusable principle holds: keep the analytical logic region-neutral and make language, folder mapping, metadata localization, and format conversion explicit in the packaging layer (CDE 2020).

IMPLEMENTATION LESSONS: REVIEWABILITY BY DESIGN

The main challenges were reproducibility and portability, not statistical methodology. Operating-system and R-version differences can affect installation and warnings, and qualifying packages at scale requires a controlled, risk-based framework. What made these manageable was designing for reviewability from the start: every control in the two submissions was chosen to leave evidence a reviewer could use. Table 2 links each design pattern to the evidence it produced.

Design Pattern	Evidence Produced for Review
Hybrid SAS/R boundary; derivations retained in ADaM	Traceable link between standards-compliant analysis data and R-generated outputs
Qualified open-source packages and controlled releases	Package inventory, fixed versions, qualification records, and inspectable source code
Recognizable program pattern and externalized metadata	Inspectable logic, predictable program structure, and reusable metadata files
Cross-language verification and retained logs	Comparison results, statistical checks, session information, and discrepancy dispositions
Reproduction-grade ADRG	Program inventory and executable reviewer instructions
Configurable packaging and regional layer	Explicit file, language, folder, and format decisions without changing the analytical core

Table 2. Reusable Design Patterns and the Evidence They Produced for Review

Together, these patterns form a single evidence chain: intent, execution, verification, and reviewer reproduction, that a reviewer can follow from any output back to its source.

PART II. EMERGING AI OPPORTUNITIES ACROSS THE SUBMISSION LIFECYCLE

A LIFECYCLE-BASED FRAMEWORK

Part II is organized around three practical questions of each lifecycle stage: where assistance could reduce manual effort, which mechanism fits the task, and what evidence is required before acceptance. The controlled artifacts from Part I serve as both the inputs AI works from and the sources against which its output is checked. Most opportunities are operational where an AI contribution could affect a submitted result or its interpretation, the required evidence and review should scale with that impact (FDA 2025b).

Figure 3 maps these opportunities onto the existing submission process instead of introducing a separate AI lifecycle. The outer stages are where AI may generate or summarize a candidate; the center is the acceptance model that does not change. The dashed arrow denotes learning between filings: accepted findings, reusable assets, and standards decisions may inform the next study or submission.

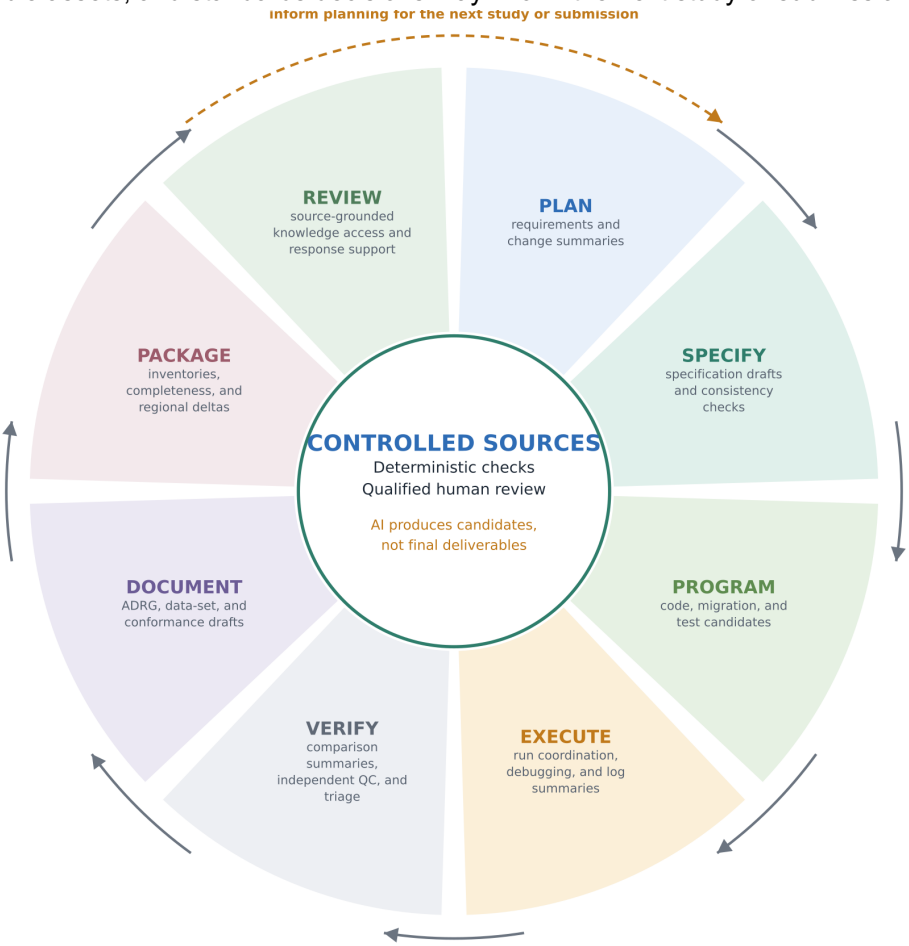

Figure 3. AI Opportunities Mapped onto a Controlled Submission Lifecycle

The stages below describe where assistance fits; the mechanism that should deliver it, and the evidence that should gate it, follow in Figure 4 and Table 3.

Plan, Specify, and Standardize

Planning and specification work is a constant reconciliation among the protocol, SAP, shells, standards, metadata, and prior decisions. It is fertile ground for assistance: an assistant could extract analysis requirements, identify missing links, summarize document changes, draft a TLF inventory, or prepare an initial programming specification. A structured intermediate record of endpoints, populations, visits, methods, variables, and output identifiers allows the accepted intent to support later programming, testing, traceability, and documentation. Standards and metadata remain authoritative. AI may flag

inconsistencies or draft derivation text, but deterministic conformance checks and standards review remain the acceptance mechanisms.

Program, Execute, Verify, and Reconcile

At the programming stage, an assistant could draft new R programs directly from approved specifications using approved packages and the recognizable program patterns, explain existing SAS or R logic, and propose unit tests or an independent implementation route. The accepted artifact remains a reviewed program that executes in the validated environment.

During execution, AI could organize warnings, classify recurring log messages, identify likely causes, and summarize failed runs for programmer review: an extension of the rules-based log-triage that already shortens review time. In QC, it could generate edge cases, propose an independent implementation route, or summarize a large comparison report; an independently AI-generated program is a natural parallel-programming analogue, and its outputs can be checked against the statistical review comments they must satisfy. Deterministic programs still perform the comparisons and produce the pass/fail evidence, and qualified reviewers close the findings.

Document, Package, and Localize

Documentation is a strong near-term opportunity because the underlying facts already exist in controlled files, and the deliverable is reviewed before submission. From sources such as the protocol, SAP, define.xml, annotated CRF, program headers, and prior templates, AI could draft package descriptions, data-set narratives, analysis explanations, and conformance text, then cross-check that names, versions, paths, and references agree. The ADRG and cSDRG are a natural fit, because their sections are largely assembled by reusing these same sources, with a qualified reviewer retaining responsibility for accuracy and final approval.

The same approach extends to eCTD assembly and regional adaptation, where a common package is tailored, and often translated, for a specific health authority. AI could compare it with a regional variant, identify missing references, summarize folder or format differences, and prepare candidate localized text for review. Exact inventories and version lists, however, remain better suited to deterministic extraction or templates, so that identifiers are verified rather than generated.

Review, Learn, and Scale

A source-grounded assistant can help internal teams or reviewers locate which program produced an output, which package version was used, where a variable was defined, or which ADRG instruction applies to a file, with every answer pointing back, through governed retrieval interfaces, to the approved source behind it. Accepted findings, approved text, validation results, and reusable assets then feed the next study or submission.

Scaling beyond individual use is a governance problem before it is a modeling one: it requires approved data boundaries, controlled retrieval, evaluation sets, model and workflow versioning, execution logging, change management, and fallback procedures.

SELECTING THE APPROPRIATE MECHANISM

Not every task needs a generative model, and some judgements still require human expertise that AI cannot replace. Figure 4 classifies the lifecycle examples by two axes: the availability of authoritative, structured evidence, and the amount of clinical, statistical, or regulatory judgment required. Exact facts belong to deterministic extraction; routine formatting and assembly to conventional automation; evidence-rich synthesis to source-grounded AI; and analysis choices, clinical interpretation, and final dispositions to qualified experts.

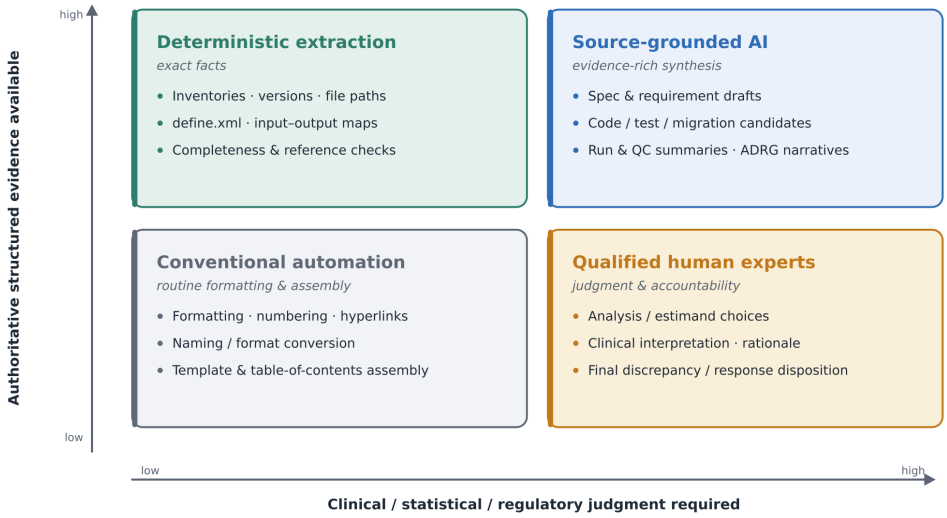

Figure 4. Selecting the Automation Mechanism by Available Structured Evidence and Required Judgment

The three elements are therefore complementary: Figure 3 shows where assistance may fit, Figure 4 which mechanism should deliver it, and Table 3 the minimum evidence and accountable role for acceptance.

REPRESENTATIVE TASKS AND ACCEPTANCE EVIDENCE

Mechanism	Lifecycle Examples	Minimum Acceptance Evidence
Deterministic extraction	Inventories; versions/paths; define.xml; input–output mappings; completeness and reference checks	Exact source comparison; reproducible extraction log
Conventional automation	Formatting/numbering; hyperlinks; naming/conversion; template and TOC assembly	Rule/template execution record; output check
Source-grounded AI	Requirement/specification drafts; code/test/migration candidates; run/QC summaries; ADRG, localized, and review-support drafts	Source anchors and contradiction checks; code/tests execute and pass independent QC; qualified approval
Human-led	Analysis/estimand choices; clinical interpretation; unusual rationale; final discrepancy or response disposition	Qualified-expert decision record; AI is not the acceptance authority

Table 3. Representative AI-Assisted Tasks and Minimum Acceptance Evidence

WORKED EXAMPLE: AI-ASSISTED ADRG AUTHORING

The ADRG is a useful worked example because different sections call for different mechanisms, and reviewer's-guide authoring is a realistic near-term target. Exact program, package, version, path, and reference information is best extracted deterministically; template structure and tables of contents suit conventional automation; source-grounded AI can populate or update narratives and flag contradictions; and qualified reviewers resolve ambiguities and approve the document. Current-study sources such as SAP, define.xml, program inventories, package records, and environment files stay authoritative, while

approved templates and prior ADRGs supply reusable structure and precedent language. Figure 5 shows the resulting sequence and the mechanism responsible at each step.

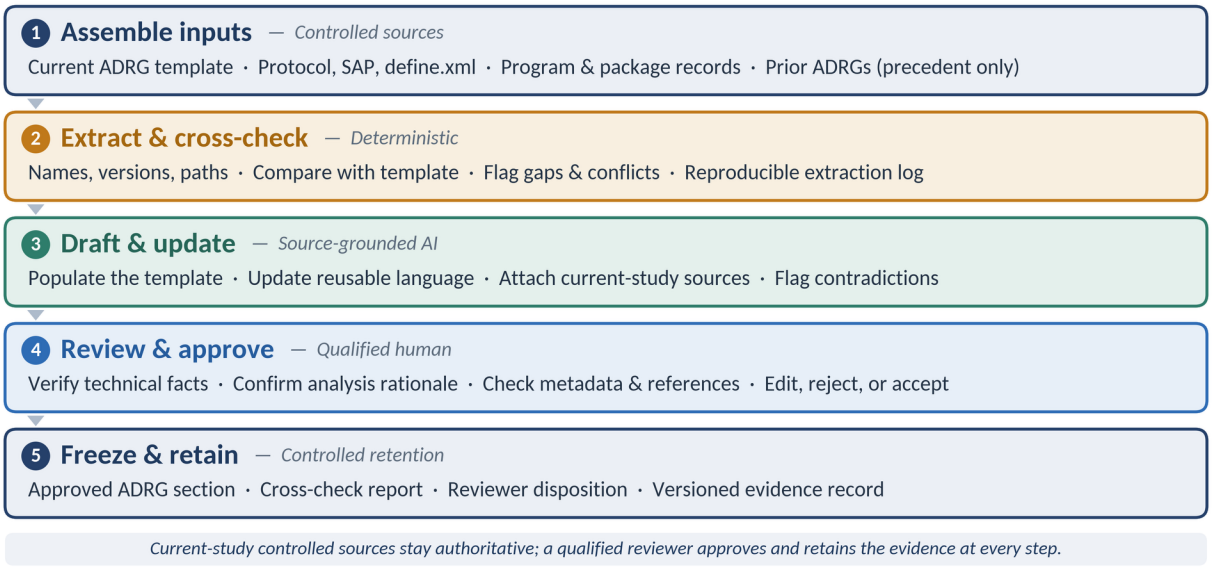

Figure 5. AI-assisted ADRG Drafting and Consistency Review, with the Accountable Mechanism at Each Step

GOVERNANCE AND PRACTICAL ADOPTION

AI-assisted work should extend existing submission controls, not replace them; a qualified human stays in the loop throughout. Four controls are essential:

- **Traceability.** Retain the approved source set, model and workflow version, generated candidate, automated checks, reviewer edits, and final disposition.
- **Reproducibility.** Attach the guarantee to the reviewed, frozen, version-controlled artifact and its evidence packet, not to the expectation that a generative model will reproduce identical wording.
- **Human accountability.** A qualified programmer, statistician, standards expert, clinician, or submission lead remains responsible for acceptance, with review depth proportional to risk.

Information protection and change control. Keep patient-level data, proprietary code, and confidential submission content within approved boundaries; changes to models, prompts, retrieval logic, or platforms require proportionate testing and release control.

Table 4 describes maturity, not a calendar. Teams should enter at the level supported by their controlled sources, validation evidence, security boundaries, and change-control capability can support.

Adoption Level	Focus	Representative Opportunities	Required Foundation
Start	Extract and check	Inventories; exact reference and cross-document checks; change and log summaries	Controlled sources; deterministic rules; reproducible logs; named reviewer
Expand	Draft and assist	ADRG/specification/localized drafts; test cases; program scaffolds	Source grounding; evaluation set; execution/checking evidence; qualified approval

Adoption Level	Focus	Representative Opportunities	Required Foundation
Scale	Integrate across workflows	Knowledge assistants; broader code/QC generation; multi-step workflows	Approved data boundaries; versioning; change control; monitoring; fallback

Table 4. Adoption Levels for AI-assisted Submission Work

CONCLUSION

The J&J experience shows that scalable R submissions depend on reviewability by design. The hybrid SAS/R boundary preserved an established ADaM layer while enabling R reporting at production scale; qualified open-source packages and recognizable programs kept the environment inspectable; independent verification produced durable evidence; the ADRG enabled reviewer reproduction; and packaging decisions were isolated from the analytical core so the operating model could be adapted across filings and regions.

The same evidence chain provides a practical basis for AI assistance. The lifecycle view identifies where AI may help, the evidence-and-judgment matrix determines the appropriate mechanism, and acceptance evidence keeps responsibility with qualified people. The practical near-term focus is bounded assistance for tasks with controlled sources and explicit acceptance evidence, while existing submission controls continue to define acceptance and accountability.

REFERENCES

Center for Drug Evaluation (CDE), National Medical Products Administration. 2020. "Guideline for Clinical Trial Data Submission (Trial)" [药物临床试验数据递交指导原则（试行）]. Beijing: CDE. Available at https://www.nmpa.gov.cn/xxgk/ggtg/ypggtg/ypqtggtg/20200720171201514.html. Accessed June 30, 2026.

Food and Drug Administration. 2025a. "Specifications for File Format Types Using eCTD Specifications, Version 9.3." August 20, 2025. Available at https://www.fda.gov/media/85816/download. Accessed June 30, 2026.

Food and Drug Administration. 2025b. "Considerations for the Use of Artificial Intelligence to Support Regulatory Decision-Making for Drug and Biological Products." Draft Guidance. January 2025. Available at https://www.fda.gov/media/184830/download. Accessed June 30, 2026.

Liu, H. 2026. "AI in Regulatory Submissions: Health Authority Expectations and Emerging Opportunities for Data Standards and Submission." Proceedings of PharmaSUG China 2026.

R Consortium R Submissions Working Group. 2023. "R Submissions Pilot 3." Available at https://github.com/RConsortium/submissions-pilot3-adam-to-fda. Accessed June 30, 2026.

R Consortium R Submissions Working Group. 2026. "R Submissions Working Group: 2026 Plans and 2025 Success." Available at https://r-consortium.org/posts/submissions-wg-2026/. Accessed June 30, 2026.

ACKNOWLEDGMENTS

The author thanks the colleagues who contributed to the J&J R submission implementation and the broader industry community whose work informed the AI opportunity synthesis. The AI framework and worked example in this paper are conceptual and do not represent a production-validated J&J system. The views expressed are those of the author and do not necessarily reflect the views or policies of Johnson & Johnson, any affiliated company, or any health authority.

RECOMMENDED READING

- *J&J: A Hybrid SAS/R Submission Story. R Consortium R Adoption Series, 2025. https://r-consortium.org/webinars/jnj-hybrid-sas-r-submission-story.html.*
- *Johnson & Johnson's Hybrid R Submission Strategy: Successful Story of the R Submission to the FDA. PHUSE SDE 2025, West Windsor, NJ. https://phuse.s3.eu-central-1.amazonaws.com/Archive/2025/SDE/US/West%20Windsor/PRE_WestWindsor03.pdf.*
- *How Can We Ensure That Health Authorities Reproduce the Sponsor's R-Based Submission Results? China PharmaRUG 2025. https://pharmarug.github.io/pharmarug-2025/program.html.*
- *From Pilot to Practice: Navigating the New Era of R-Based Regulatory Submissions. China PharmaRUG 2026. https://pharmarug.github.io/pharmarug-2026/program.html.*
- *PharmaSUG 2026 US Conference Proceedings. https://pharmasug.org/conferences/pharmasug-2026-us/conference-proceedings/*
- *R Validation Hub: risk-based package assessment resources. https://www.pharmar.org*
- *Pharmaverse: open-source R packages for clinical reporting. https://pharmaverse.org*

CONTACT INFORMATION

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

Wenlan Pan
Johnson & Johnson
+86 18806160718
wpan16@its.jnj.com