



Transitioning to R for Clinical Submissions

twitter:@rinpharma github:philbowsher

A scenic landscape photograph showing a winding dirt path through a grassy field with patches of green vegetation. In the background, a calm lake is visible, surrounded by a dense forest of green trees. The sky is overcast with soft, grey clouds. The image is partially obscured by a white diagonal shape on the right side, which serves as a background for the text.

Focus early on package management.
Consider hiring an R admin to help manage
the package environment for clinical
reporting.



Step 1

- **Add RStudio as a Trusted Vendor**
- This will speed things up considerable
- This will add RStudio software and packages as trusted by IT & QA
- [Helpful validation documents are here](#)
- Verification docs: [RSC](#), [RSP](#), [RSPM](#), [RSPM2](#)



Step 2

- **Get users off of desktop**
- If you must use desktops, still follow the advice in the steps below
- Desktop can be equally challenging for IT and R users



Step 3

- **Get a Server (cloud or onsite) for Development**
- IT will manage the server for users
- Have users use RStudio Projects, renv and version control
- RStudio IDE has integrated support for version control



Step 4

- **Create an R version and Package plan**
- Which version(s) of R will be supported?
- How often will you update packages?
- Which packages will be validated?
- [Package Strategy Maps](#)



Step 5

- **Created a Validated repository of packages based on Plan in Step 4**
- [RStudio Validation Docs](#)
- [Add other packages according to your RISK-BASED APPROACH](#)
- This is your Shared Baseline
- Freeze repo when you upgrade packages or add R versions
- IT can set repo as default via Renviron.site for users



Step 5 Info

- IT can surface out the validated packages in RStudio via the System Library
- This will prevent the need for users to install packages
- Alternatively, Users can install packages from the Validated repo
- renv can be used to isolate the packages for the project users are working on
- An R package library is tied to a specific version of R, so different R versions have distinct package libraries



Step 5.5

- **As time goes on, Don't upgrade R**
- Rather, Install multiple versions of R on the same server
- [R Installs](#)
- Point older versions of R to frozen repo states
- Don't install packages from CRAN
- Organizations will have multiple repositories (CRAN, validation etc.) aligning to different use cases



Steps 1-5 are about managing packages and R. Steps 6-10 are about internal processes for your own code, packages and infrastructure.



Step 6

- **Validate your internal packages & UDFs**
- Ellis Hughes
- <https://rinpharma.com/publication/rinphar>



Step 7

- **Determine your method for data management & TLFs**
- RStudio projects, tidyverse, htmlwidgets, RMarkdown, GT
- <https://db.rstudio.com/>
- Roche way, rtables, loon
- GSK way, pharmartf, timber, xportr, matacore
- Merck way, r2rtf, pkglite, cleanslate
- Others, officer?



Step 8

- **Harden Production and not RStudio**
- When reporting is done, migrate the processes to a production environment
- Learn basic software development practices like testing, CI/CD, automation tools, versioning (git) etc.
- Create Shiny apps to standardize reporting and enable new capabilities
- [The future of clinical tf1s](#)



Step 9

- **Create a R cloud strategy for clinical reporting**
- [RStudio Cloud tools](#) for VPCs
[cloudformation](#) & [ARM](#)
- Review Kubernetes as a container-orchestration system
- Consider Validating Docker Images
- If you are installing packages as part of a Dockerfile, it is important to make sure you are setting the correct frozen repo in the `install.packages` command



Step 10

- **Listen at Rin Pharma**
- Learn what other pharmas are doing in the DS space
- <https://rinpharma.com/>



rstudio conference & rinpharma