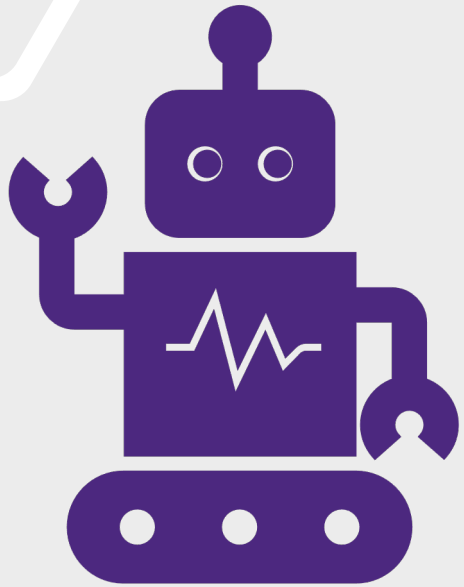
Making R work for you (with automation!)

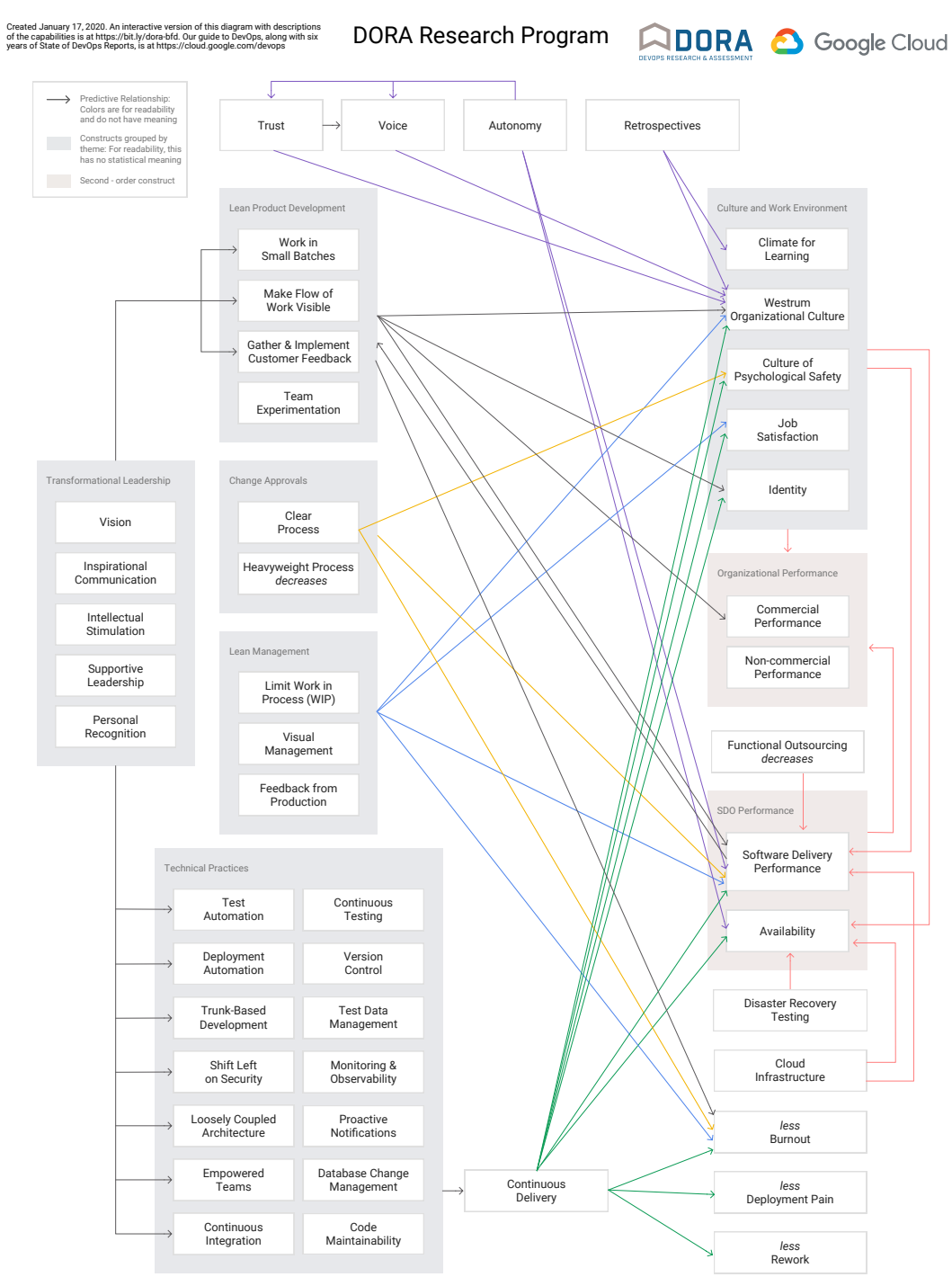
John Benninghoff

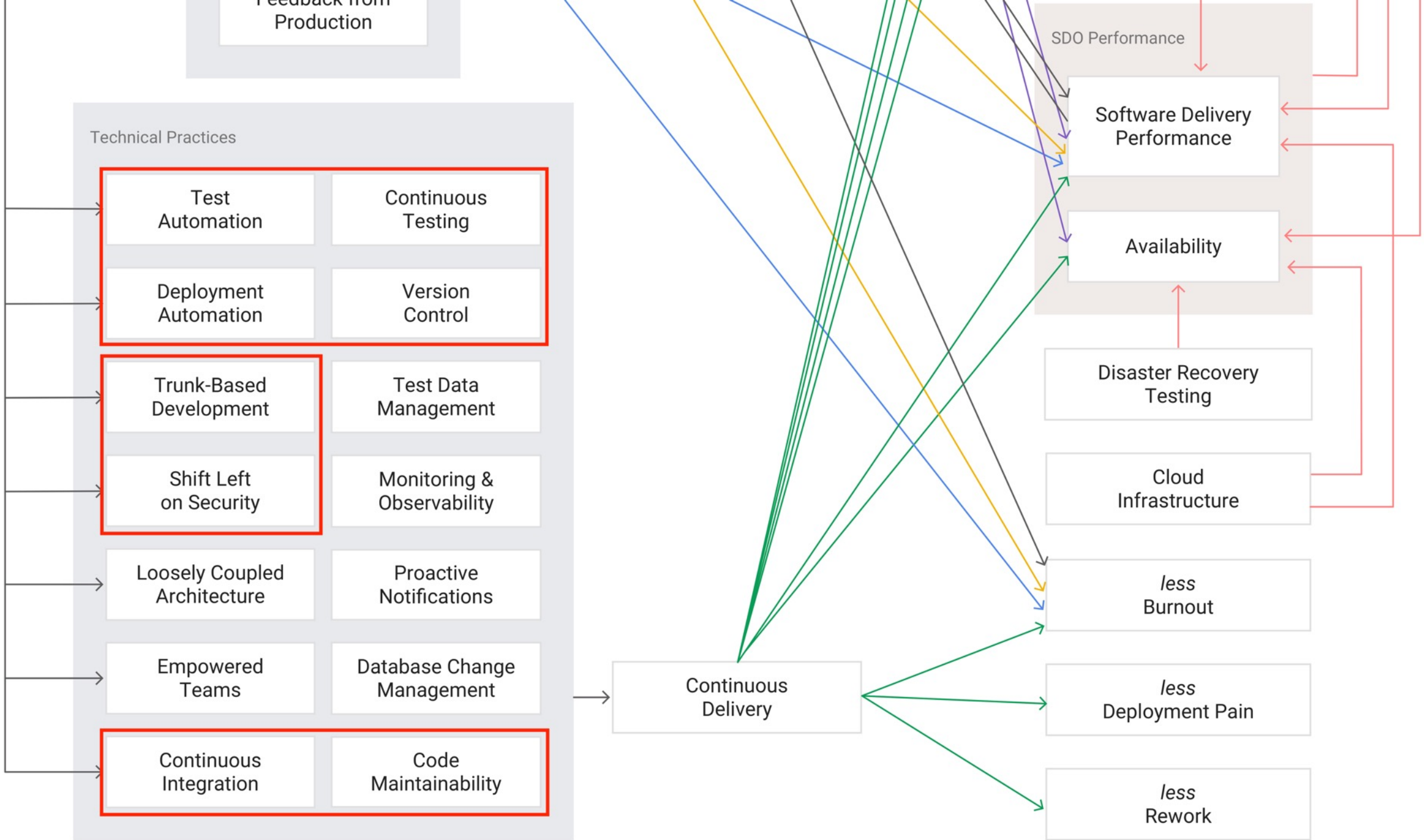


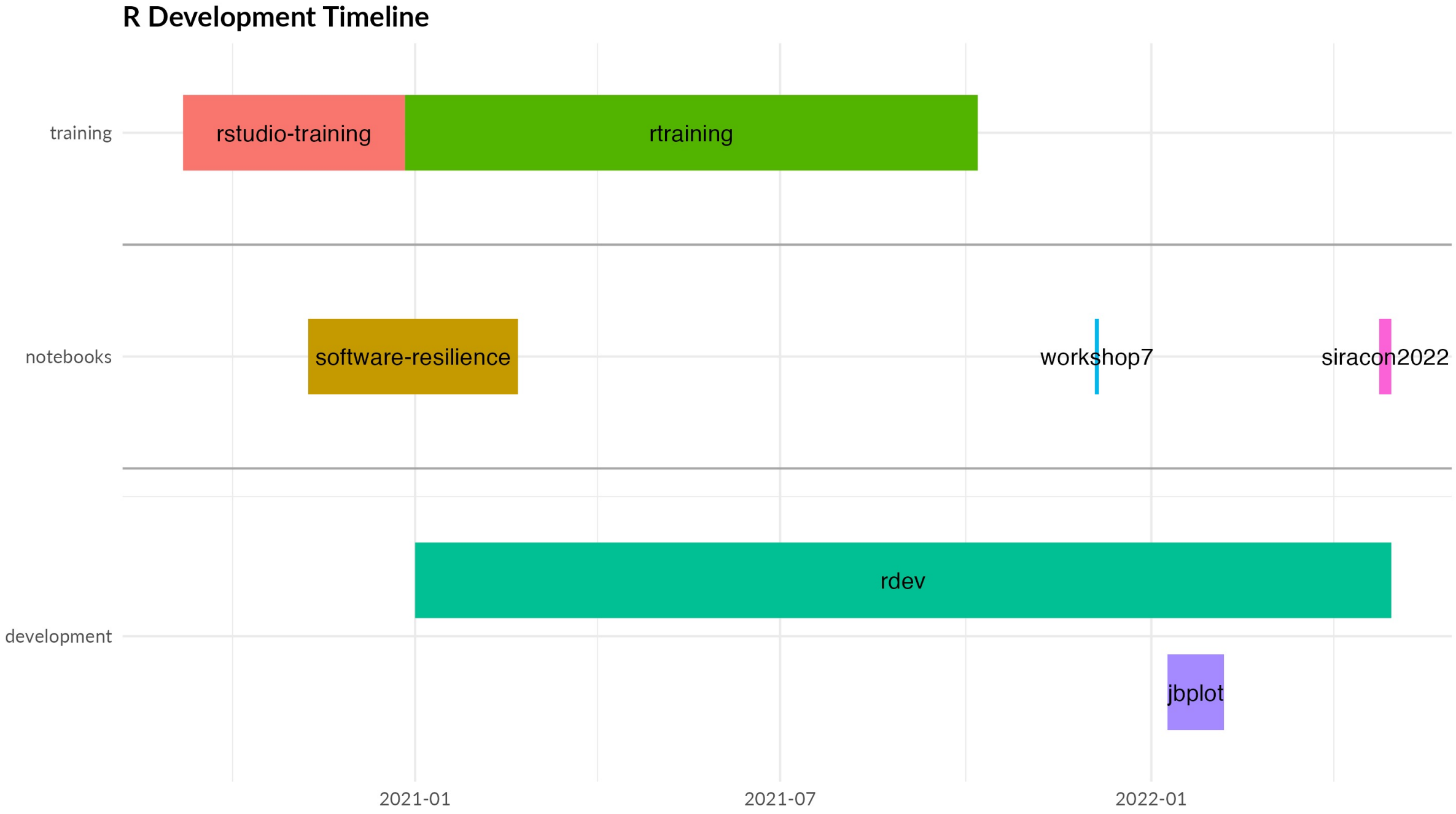

ONSite
651 · 429 · 3781
RENT ME!


ONSite
429

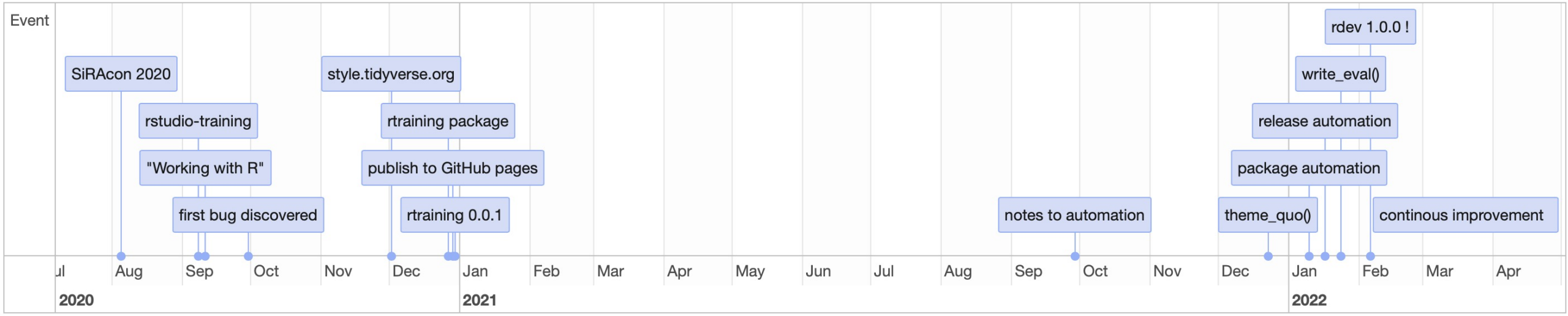
DevOps Research




Feedback from Production
Technical Practices
Test Automation
Continuous Testing
Deployment Automation
Version Control
Trunk-Based Development
Test Data Management
Shift Left on Security
Monitoring & Observability
Loosely Coupled Architecture
Proactive Notifications
Empowered Teams
Database Change Management
Continuous Integration
Code Maintainability
Continuous Delivery
SDO Performance
Software Delivery Performance
Availability
Disaster Recovery Testing
Cloud Infrastructure
less Burnout
less Deployment Pain
less Rework


R Development Timeline
training
notebooks
development
rstudio-training
rtraining
software-resilience
workshop7
siracon2022
rdev
jbplot
2021-01
2021-07
2022-01

Introduction

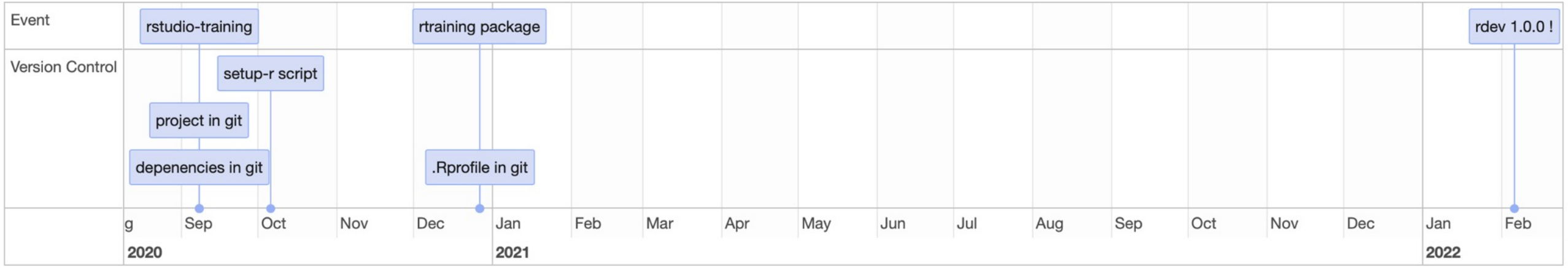


Version Control

Put everything (except artifacts) into version control for reproducibility and history.

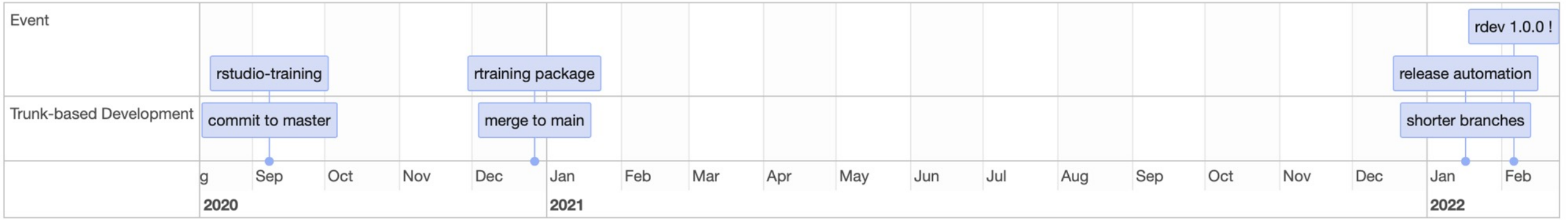
Packages

- renv: dependency management



Trunk-Based Development

Linear development avoids code conflicts.



Shifting Left on Security

Maintenance first ensures you get it done.

Packages:

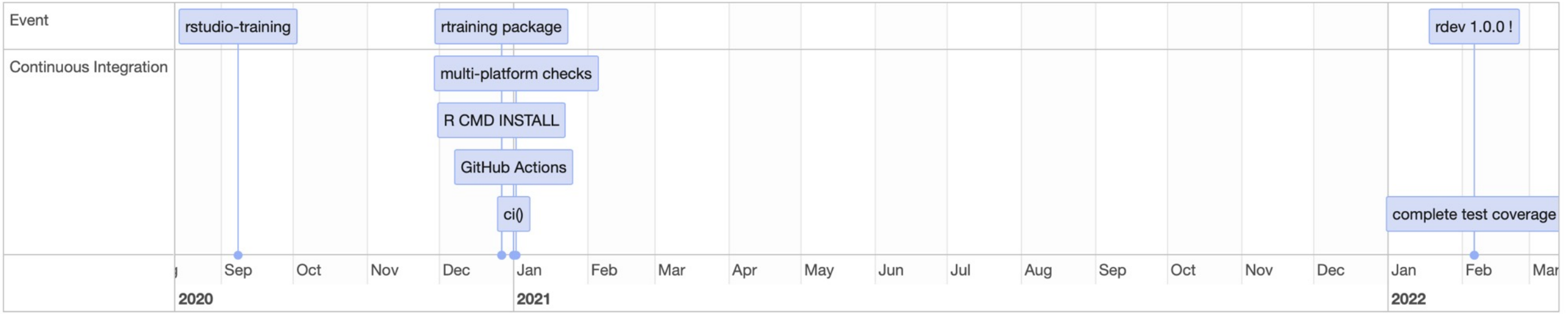
- renv

Continuous Integration

Build and test on each commit to catch mistakes early.

Packages:

- devtools
- usethis
- r-lib/actions



Local CI

Description

Run continuous integration tests locally.

Usage

```
ci(renv = TRUE, styler = NULL, lintr = TRUE, document = TRUE, rcmdcheck = TRUE)
```

Arguments

`renv`	check `renv::status()`
`styler`	style all files using `style_all()`, see details
`lintr`	lint all files using `lint_all()`
`document`	run `devtools::document()`
`rcmdcheck`	run `R CMD check` using: `rcmdcheck::rcmdcheck(args = "--no-manual", error_on = "warning")`

Details

If `renv::status()` is not synchronized, `ci()` will stop.

If `styler` is set to `NULL` (the default), `style_all()` will be run only if there are no uncommitted changes to git. Setting the value to `TRUE` or `FALSE` overrides this check.

If `lint_all()` finds any lints, `ci()` will stop and open the RStudio markers pane.

Examples

Run examples

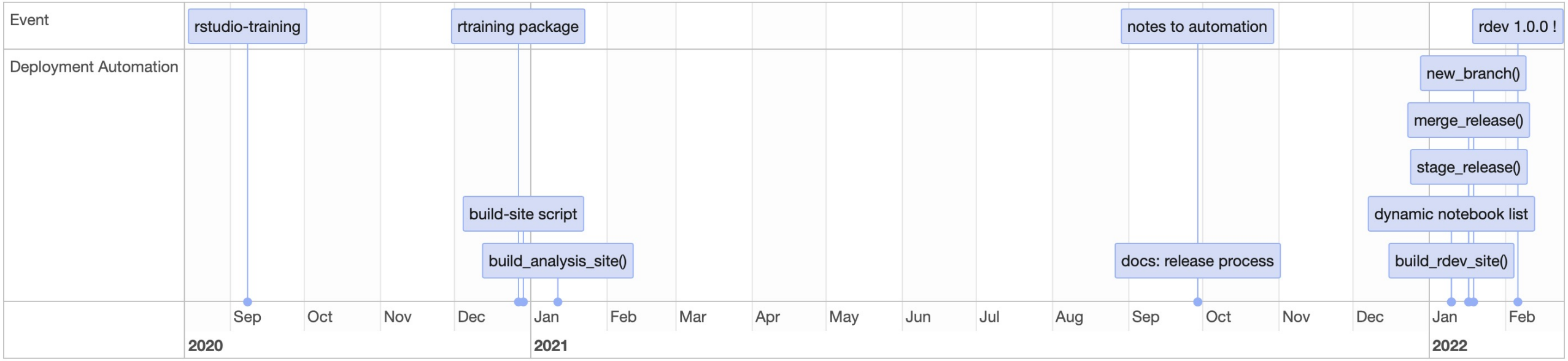
```
## Not run:
ci()
ci(styler = TRUE)
ci(styler = FALSE, rcmdcheck = FALSE)

## End(Not run)
```

Deployment Automation

Automate your development workflow to spend more time writing.

Packages:

- pkgdown, rmarkdown: build_analysis_site()
- gert, gh: git, GitHub automation



Workflow

- new_branch(): Create a new branch and bump 'dev' version to 9000
- write the code, test(), commit, ci(), repeat
- stage_release(): Open a GitHub pull request for a new release from NEWS.md. Calls build_analysis_site() or build_rdev_site() to build GitHub pages (README, notebooks, package docs)
- wait for GitHub Actions to complete successfully
- merge_release(): Merge and create a new release on GitHub.

Dynamic notebook lists

- rmd_metadata(): Extract the YAML front matter and 'description' line from an analysis notebook, and construct a URL to the notebook's location on GitHub pages.

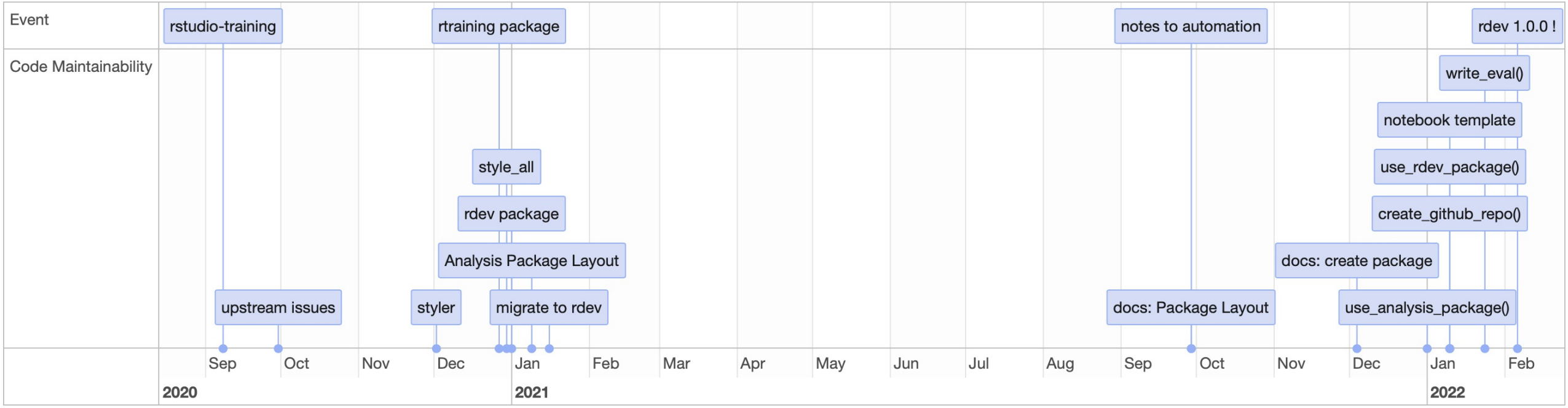
```
library(rdev)
library(fs)
library(dplyr)
library(purrr)

notebooks <- dir_ls("analysis", glob = "*.Rmd") |>
  map_dfr(rmd_metadata) |>
  mutate(bullet = paste0("- [", title, "](", url, ") (", date, "): ", description)) |>
  pull(bullet)

writeLines(notebooks)
```

Code Maintainability

Consistent and clean code is easier to understand.

Packages:

- styler
- roxygen2
- purrr
- desc



A really bad idea for maintainability

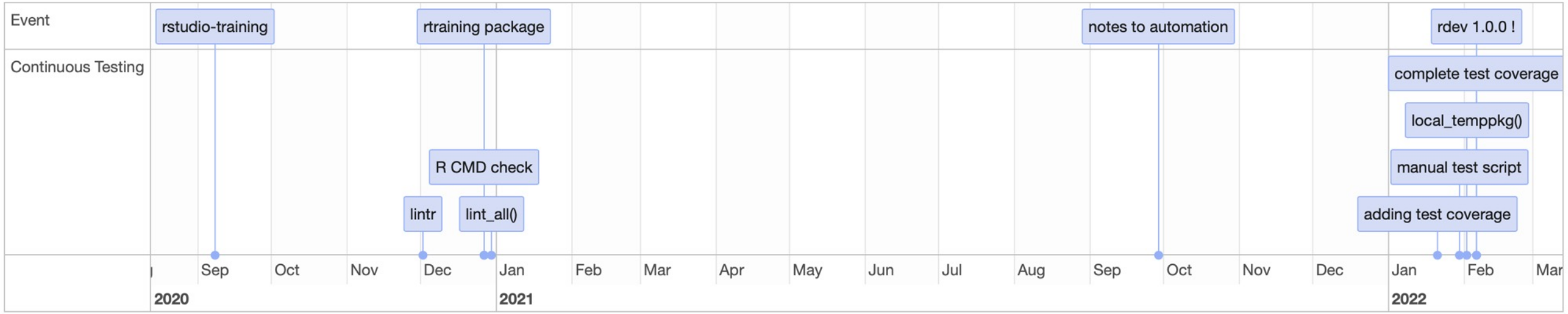
```r
#' Write and evaluate an expression
#'
#' `write_eval(string)` is a simple wrapper that prints `string` to the console using
#' [`writeLines()`][base::writeLines], then executes the expression using [`parse()`][base::parse]
#' and [`eval()`][base::eval].
#'
#' @param string An expression to be printed to the console and evaluated
#'
#' @return The return value from the evaluated expression
#'
#' @examples
#' write_eval("pi")
#'
#' write_eval("exp(1)")
#' @export
write_eval <- function(string) {
  if (!is.character(string)) stop("not a character vector")
  if (string == "") stop("nothing to evaluate")
  writeLines(string)
  eval(parse(text = string))
}
```

Continuous Testing

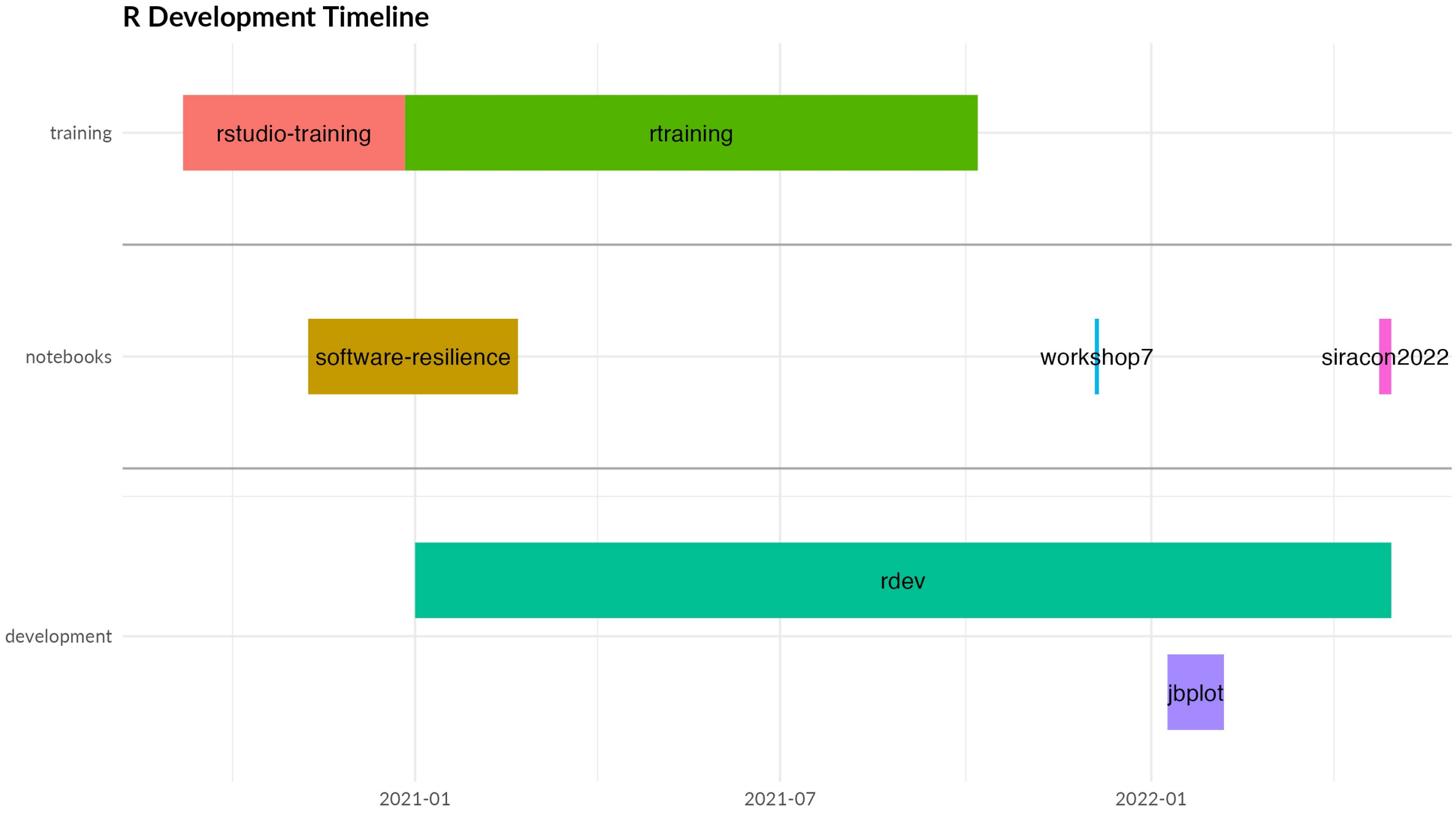
The biggest challenge: formally specifying what you are building and how it is *supposed* to work defends against the dangers of hidden assumptions.

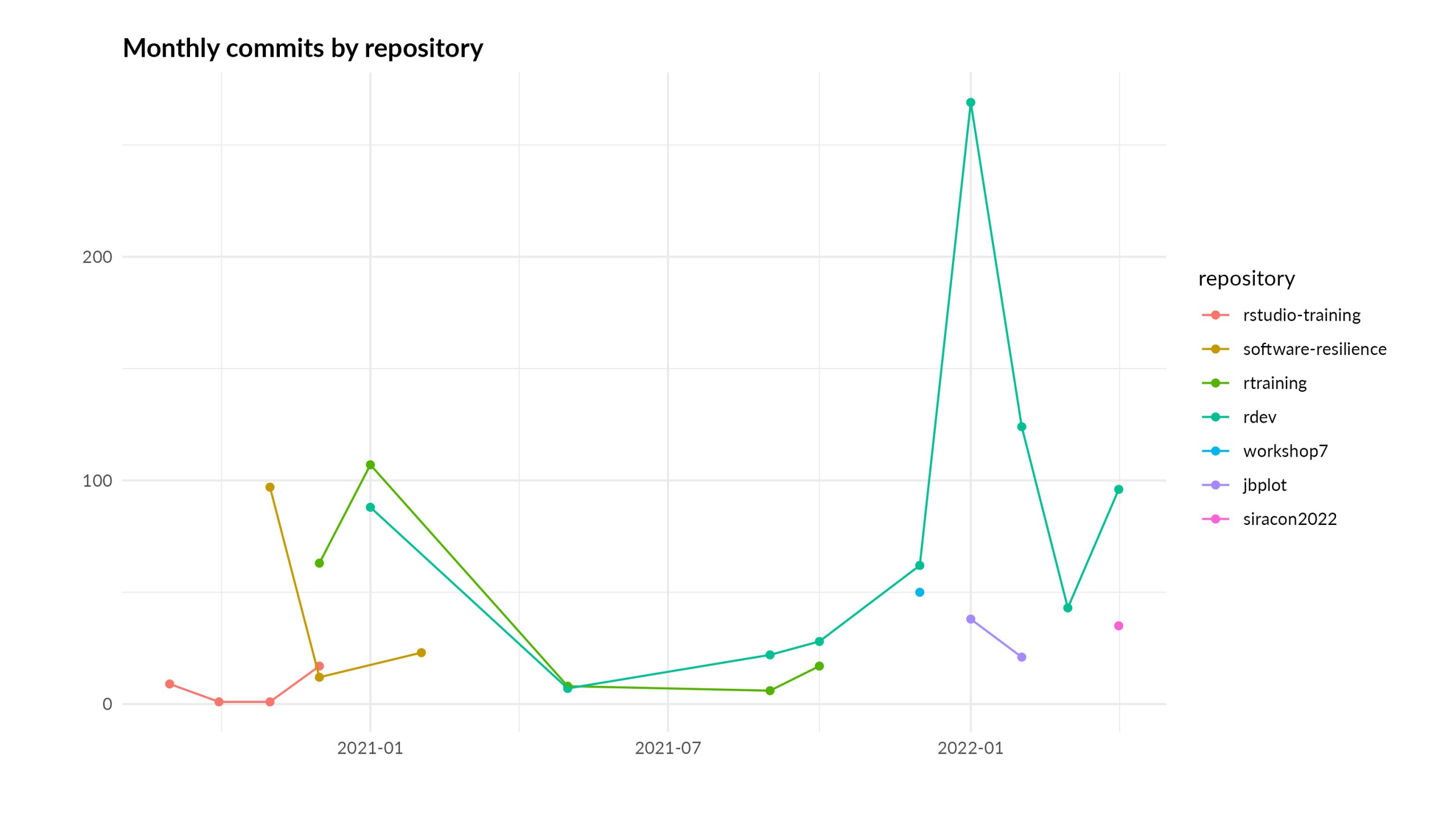
Packages

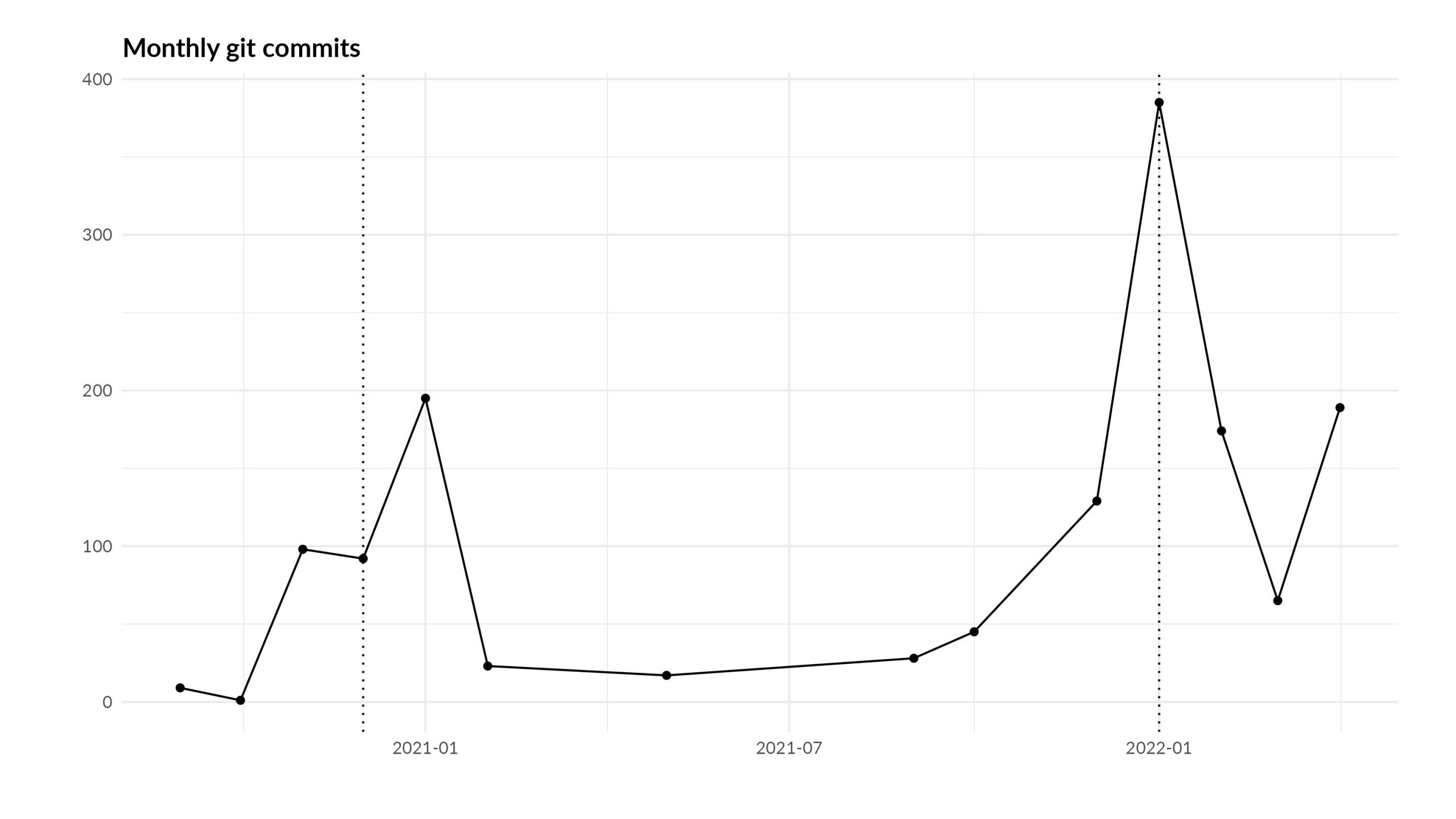
- lintr
- rcmdcheck
- testthat
- covr
- mockery
- withr
- rlang
- spelling

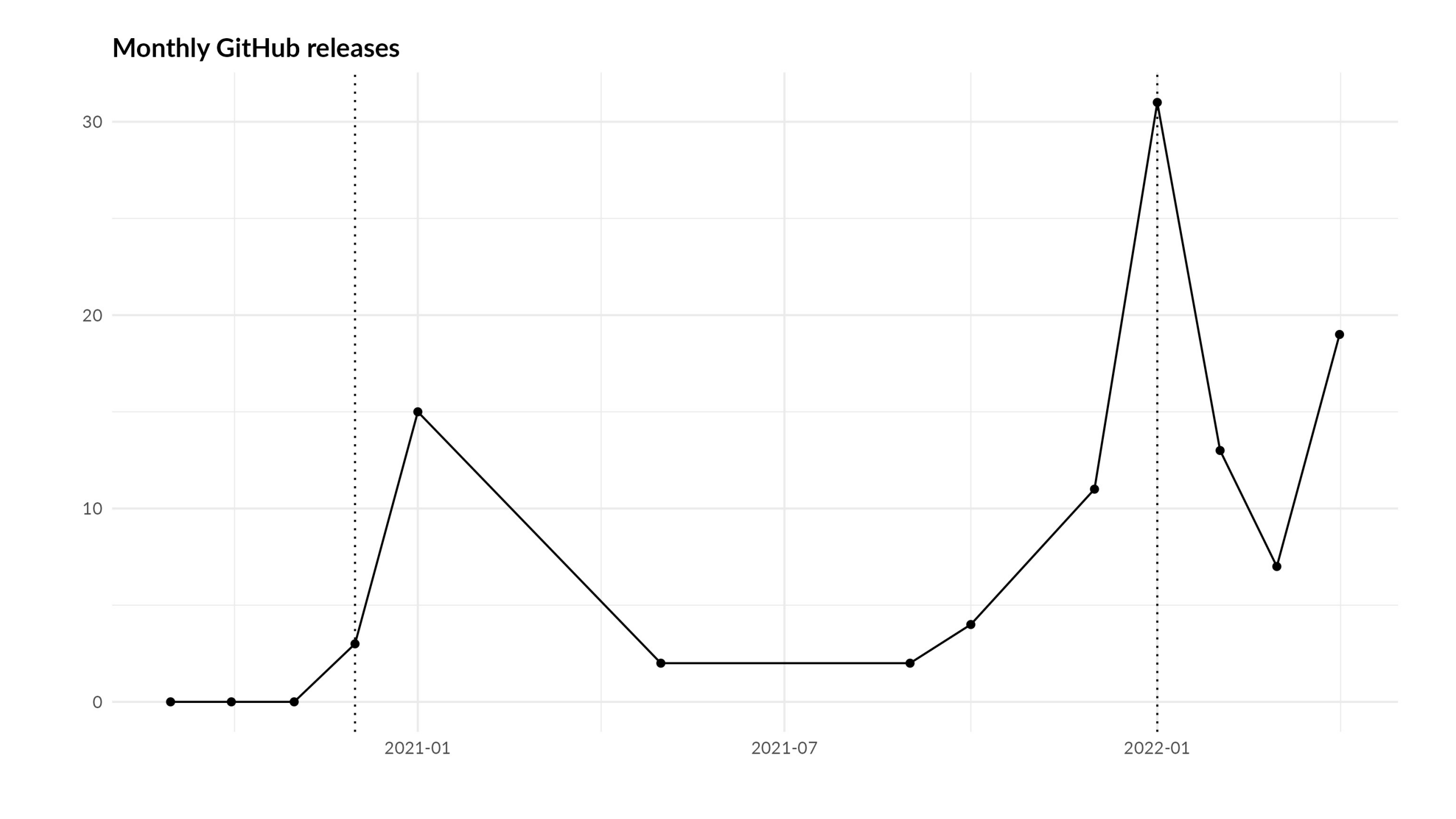


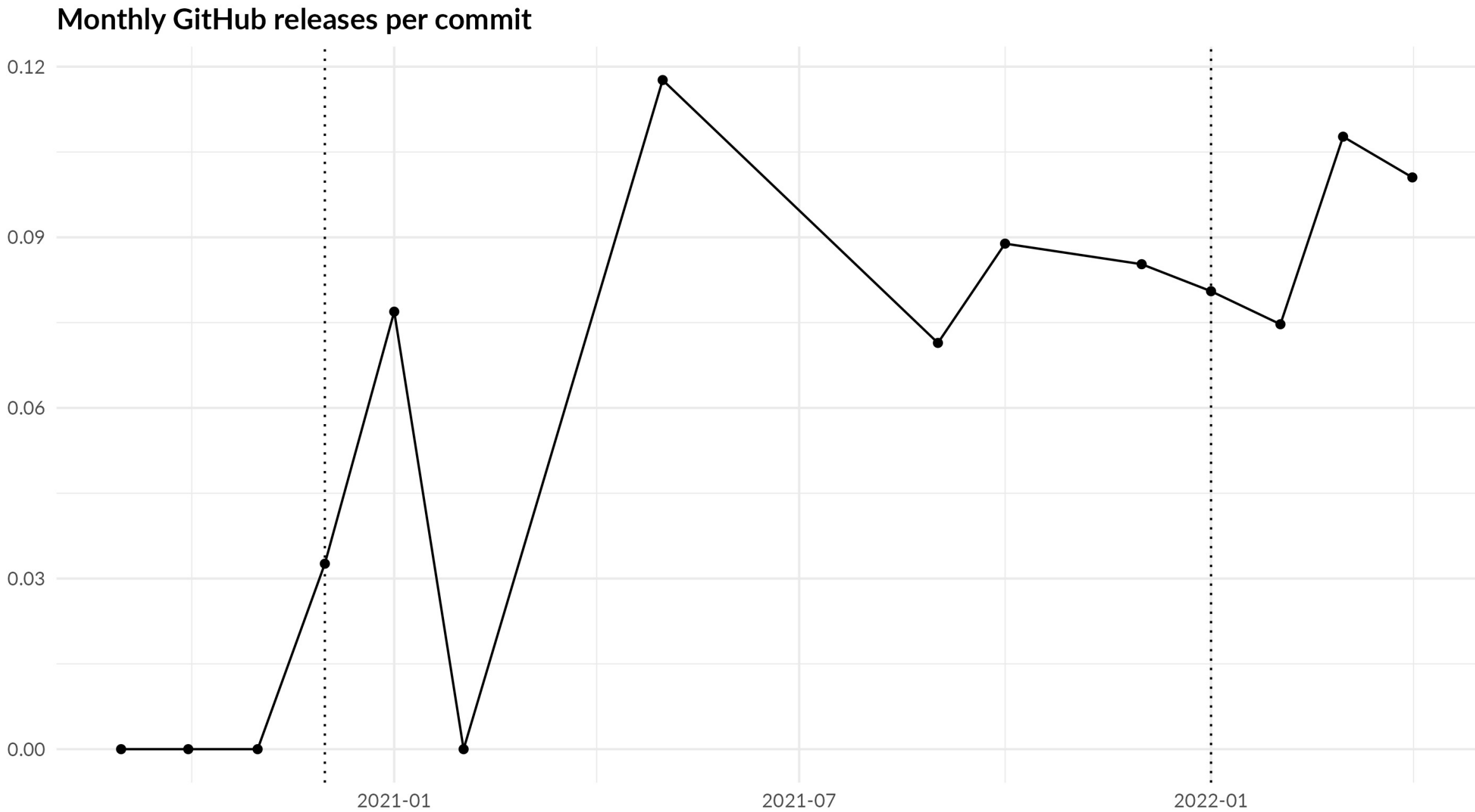
Results


R Development Timeline
training
notebooks
development
rstudio-training
rtraining
software-resilience
workshop7
siracon2022
rdev
jbplot
2021-01
2021-07
2022-01

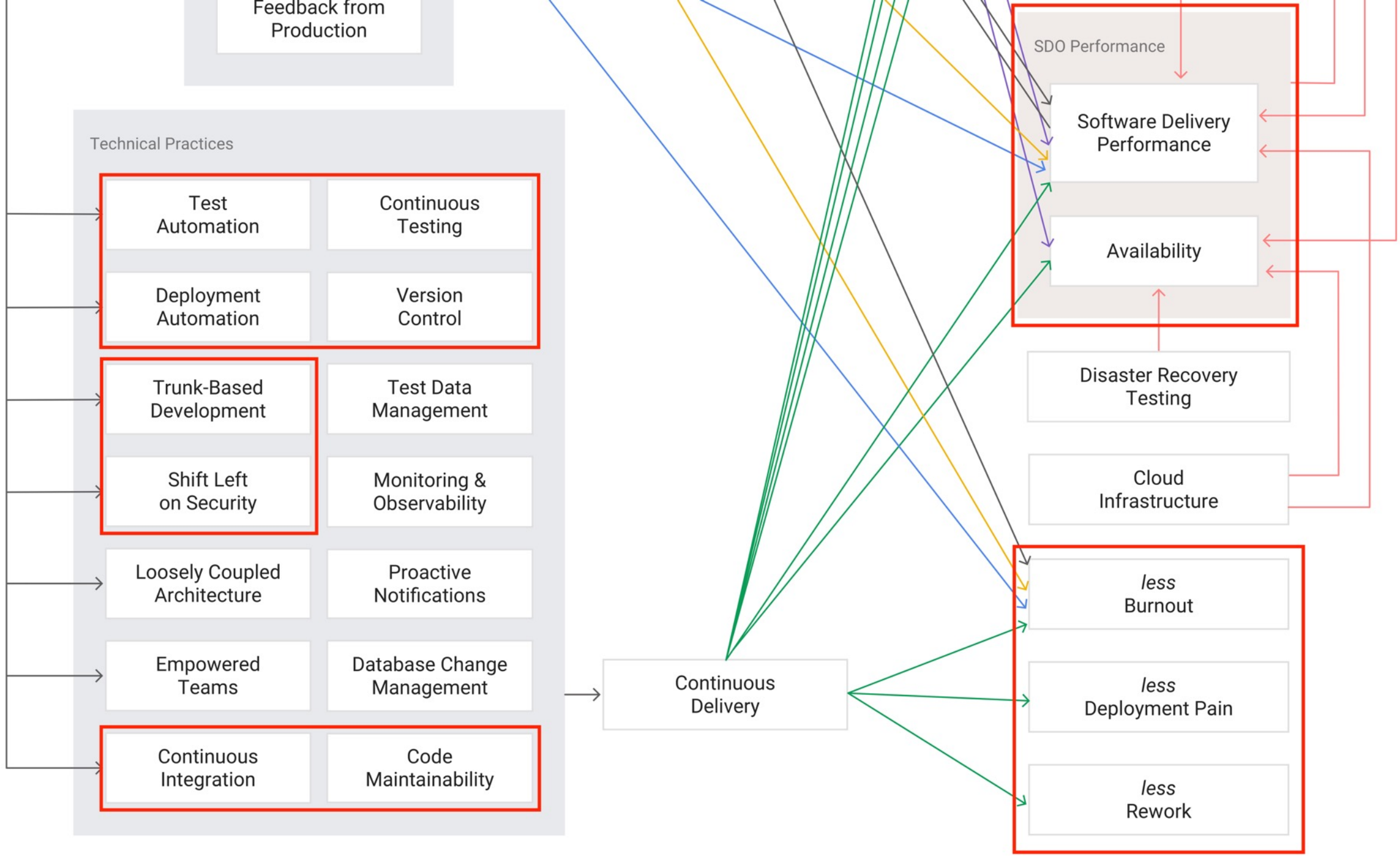

Monthly commits by repository
200
100
0
2021-01
2021-07
2022-01
repository
rstudio-training
software-resilience
rtraining
rdev
workshop7
jbplot
siracon2022


Monthly git commits
400
300
200
100
0
2021-01
2021-07
2022-01


Monthly GitHub releases
30
20
10
0
2021-01
2021-07
2022-01

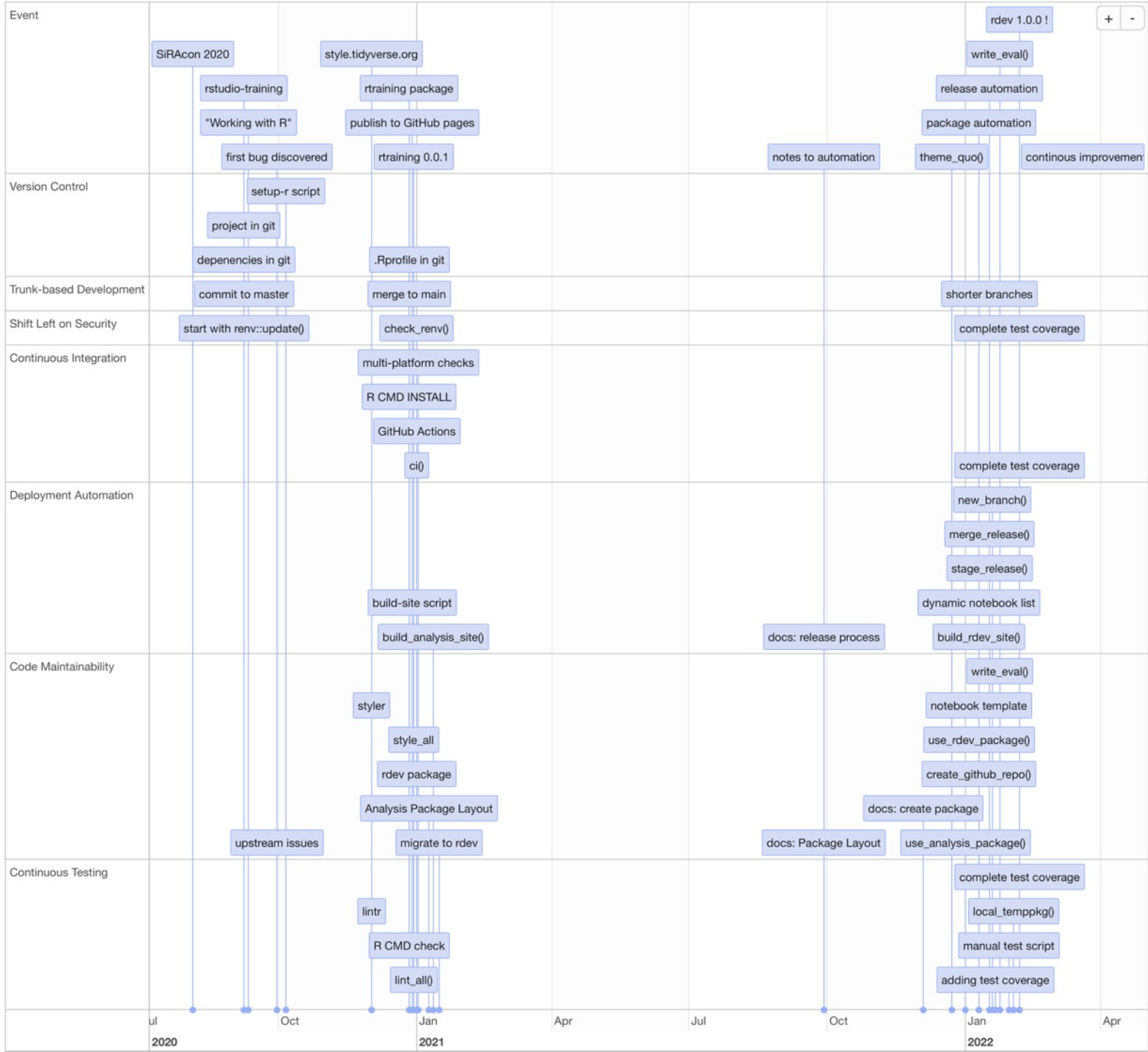

Monthly GitHub releases per commit
0.12
0.09
0.06
0.03
0.00
2021-01
2021-07
2022-01


ONSite
651 · 429 · 3781
RENT ME!


Feedback from Production
Technical Practices
Test Automation
Continuous Testing
Deployment Automation
Version Control
Trunk-Based Development
Test Data Management
Shift Left on Security
Monitoring & Observability
Loosely Coupled Architecture
Proactive Notifications
Empowered Teams
Database Change Management
Continuous Integration
Code Maintainability
Continuous Delivery
SDO Performance
Software Delivery Performance
Availability
Disaster Recovery Testing
Cloud Infrastructure
less Burnout
less Deployment Pain
less Rework

Questions?

https://www.information-safety.org
https://www.linkedin.com/in/jbenninghoff/
@jbenninghoff
jbenninghoff@mac.com


Event
SiRAcon 2020
rstudio-training
"Working with R"
first bug discovered
style.tidyverse.org
rtraining package
publish to GitHub pages
rtraining 0.0.1
rdev 1.0.0 !
write_eval()
release automation
package automation
notes to automation
theme_quo()
continous improvemen
Version Control
setup-r script
project in git
depenencies in git
.Rprofile in git
Trunk-based Development
commit to master
merge to main
shorter branches
Shift Left on Security
start with renv::update()
check_renv()
complete test coverage
Continuous Integration
multi-platform checks
R CMD INSTALL
GitHub Actions
ci()
complete test coverage
Deployment Automation
new_branch()
merge_release()
stage_release()
build-site script
dynamic notebook list
build_analysis_site()
docs: release process
build_rdev_site()
Code Maintainability
write_eval()
styler
notebook template
style_all
use_rdev_package()
rdev package
create_github_repo()
Analysis Package Layout
docs: create package
upstream issues
migrate to rdev
docs: Package Layout
use_analysis_package()
Continuous Testing
complete test coverage
lintr
local_temppkg()
R CMD check
manual test script
lint_all()
adding test coverage
ul
Oct
Jan
Apr
Jul
Oct
Jan
Apr
2020
2021
2022

rdev Packages

- desc
- devtools
- fs
- gert
- gh
- lintr
- markdown
- miniUI
- pkgdown
- purrr
- rcmdcheck
- remotes
- renv
- rlang
- rmarkdown
- styler
- tibble
- usethis
- withr
- xml2
- yaml
- covr
- DT
- knitr
- mockery
- spelling
- stringi
- testthat

Future Testing

Mutation Testing: Wikipedia

- R packages:
  - mutant
  - autotest
- Papers:
  - Does mutation testing improve testing practices?
  - Practical Mutation Testing at Scale

Formal Methods:

- Planning with flare
- Hillel Wayne
- Learn TLA+
- Alloy Documentation

References

- DORA Research
- “Working with R”
- First bug: https://github.com/rstudio/renv/issues/547
- Notebooks used to develop this presentation: siracon2022
- All my work: https://github.com/jabenninghoff