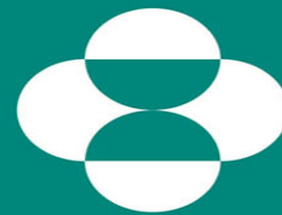


R PACKAGE ORIENTED SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) IN REGULATED CLINICAL TRIAL ENVIRONMENTS



MERCK

INVENTING FOR LIFE

March 2020

Rinki Jajoo, Yalin Zhu*, Clare Bai, Sarad Nepal,
Daniel Woodie, Keaven Anderson, Yilong Zhang*

Merck & Co., Inc., Kenilworth, NJ, USA



* Primary Author

R and R Packages Introduction

- R is a software environment for statistical computing and graphics (comparable to SAS)
 - An R function is a set of R code with arguments (comparable to SAS macro)
 - e.g. `read.csv()` is a function read a csv file into R.
 - An R package is a collection of R functions (similar to a standard SAS macros)
 - e.g. `ggplot2` is an R package to create graphs.
- R-SDLC aims to provide guidance for
 - Internal analysis tool development and validation using R
 - Build consistent programming style and principals
 - Comply with essential regulatory requirements

R Roles and Responsibilities- Cross Functional Team



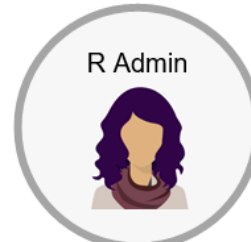
- The SME is the owner of the R-SDLC process
- Develops and maintains the R-SDLC process documentation
- Liaison between business, IT and R administrator



- Provide proof of concept to construct the specification and validation plan
- Provide requirements for development of new or enhanced R functions
- Stakeholders

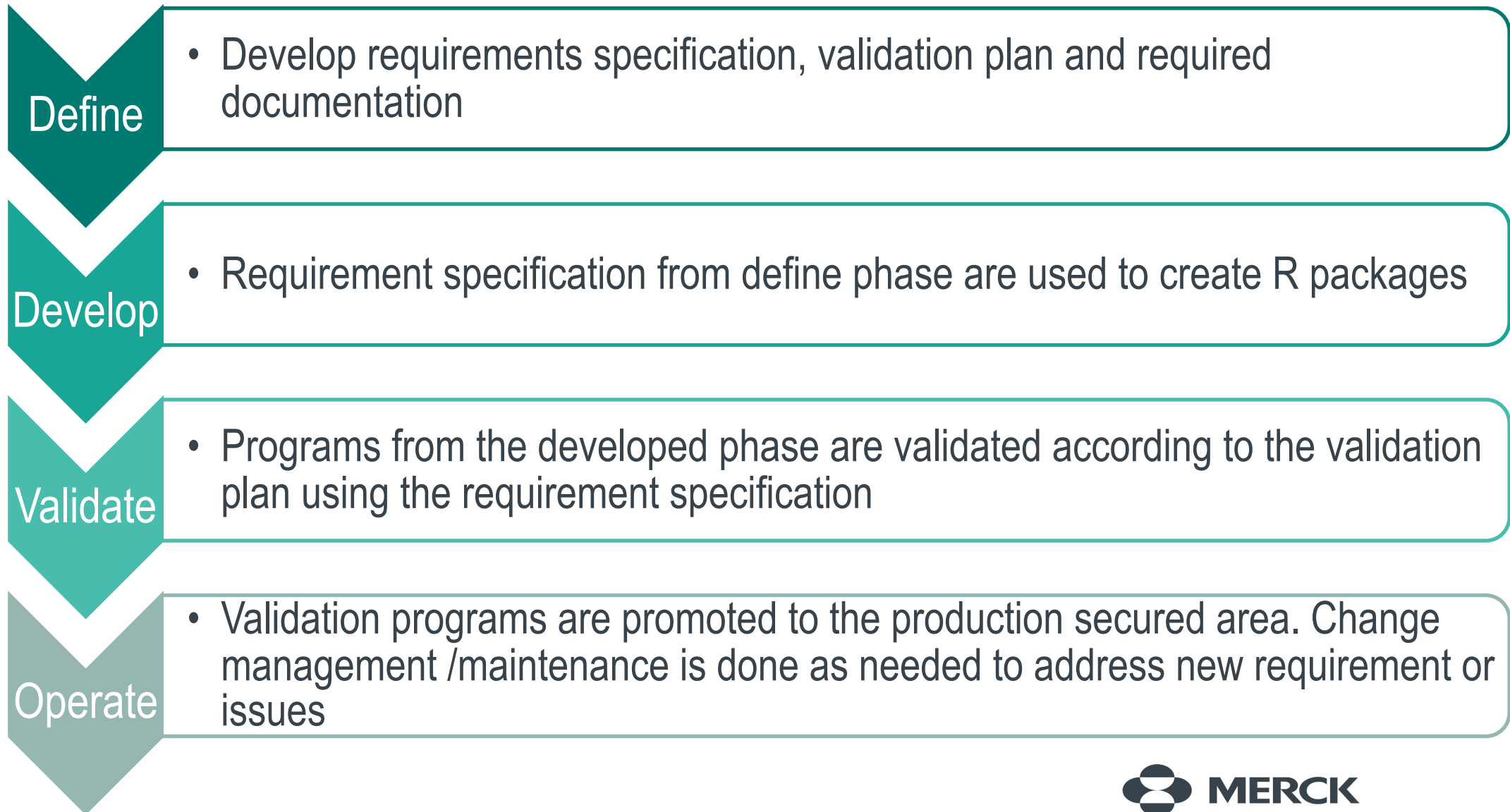


- Provide development support to Internal R packages
- Draft specification, program function and validate
- Ensure compliance of the R-SDLC process while developing an internal R package



- Maintains the internal R computing and development platform (e.g. Bitbucket and RStudio server pro, RStudio Package Manager etc.)
- Provide support to deploy, install and upgrade R packages
- Member of governance team to identify R packages in various risk categories

R programming: Software Development Life Cycle(SDLC)



R SDLC – Badge System

	Badges types
Risk categories	risk low risk medium risk open
SDLC Badges	lifecycle validation lifecycle deprecated
	lifecycle specification lifecycle development lifecycle customer review lifecycle stable lifecycle retired
	Define Development Validation Operation
Validation types (of each function):	validation independent testing validation double programming
Code coverage	coverage 0~20% coverage 20~60% coverage 60~80% coverage 80~100%
R package build status	build passing build failing

R-SDLC Cycle: Define Stage

- During the define stage programmers and statisticians initiate or update the specification of an internal R package
- Specification includes a list of functions to be developed with detailed information/logics on function input arguments and output values
- Specification can be documented using the “roxygen2” package to be displayed in PDF user manual
- Developed a standard package template for ease of use and maintaining consistency
- Standard package template provides **standard frame work** for developing standard R packages
- Standard package folder structure can be invoked by using **RStudio project template**

```
#' @section Specification:
#' \ifelse{latex}{
#'   \itemize{
#'     \item create \code{filename} function (describe
#'       your function's specification)
#'     \itemize{
#'       \item define function name (filename) and
#'         purpose
#'       \item customize filename with corresponding
#'         input and output
#'       \item link to corresponding hyperlink/url}}
#'     }
#'   The followings are some possible code:
#'   (optional)
#'   \itemize{
#'     \item \code{library(StandardPackageTemplate)}
#'     \item \code{filename("aaa")}
#'     \item \code{filename("bbb")}
#'     \item \code{filename("ccc")}
#'   }
#' }
#' }{
#' The contents of this section are shown in PDF
#' user manual only.
#' }
```

Specification

Logic for creating xxx_badge function (describe your function's specification)

- define badge name and purpose
- customize badge color with corresponding stage/purpose
- link to corresponding hyperlink/url

The followings are some possible code:

- `library(StandardPackageTemplate)`
- `risk_badge("open")`
- `sdlc_badge("specification")`
- `codecov_badge(0)`
- `build_badge("passing")`

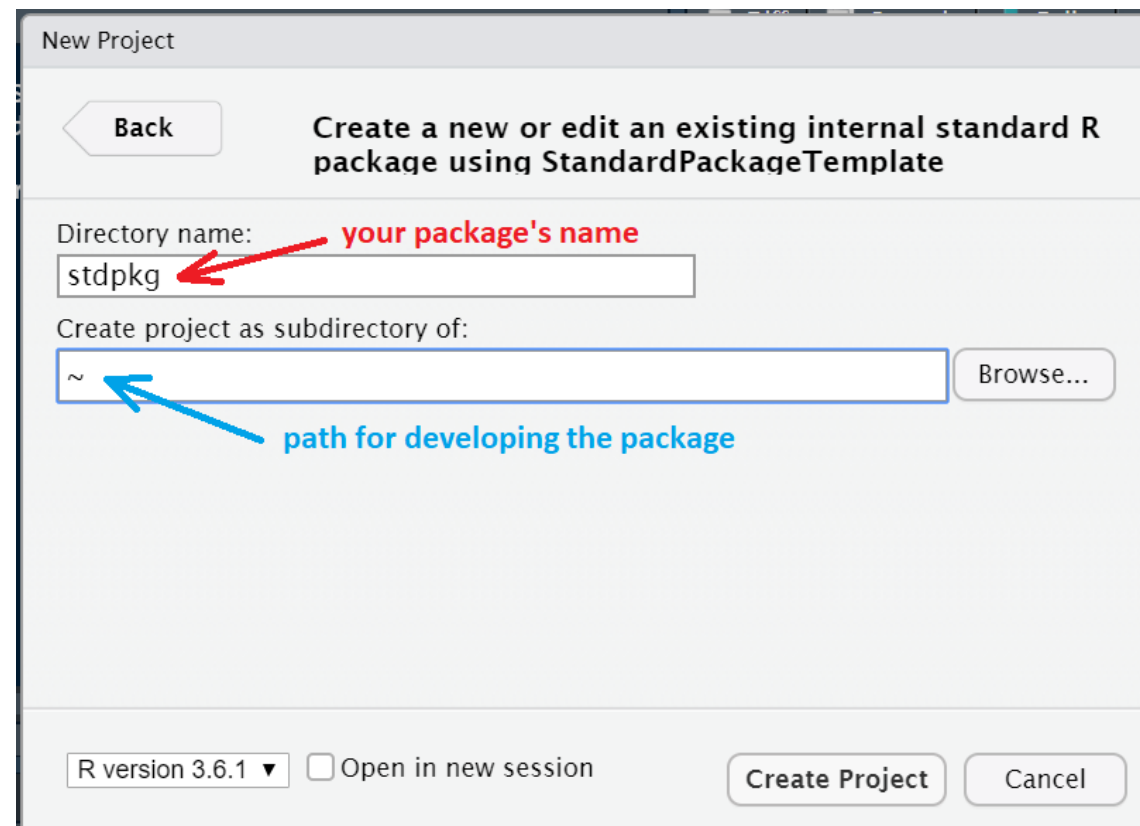
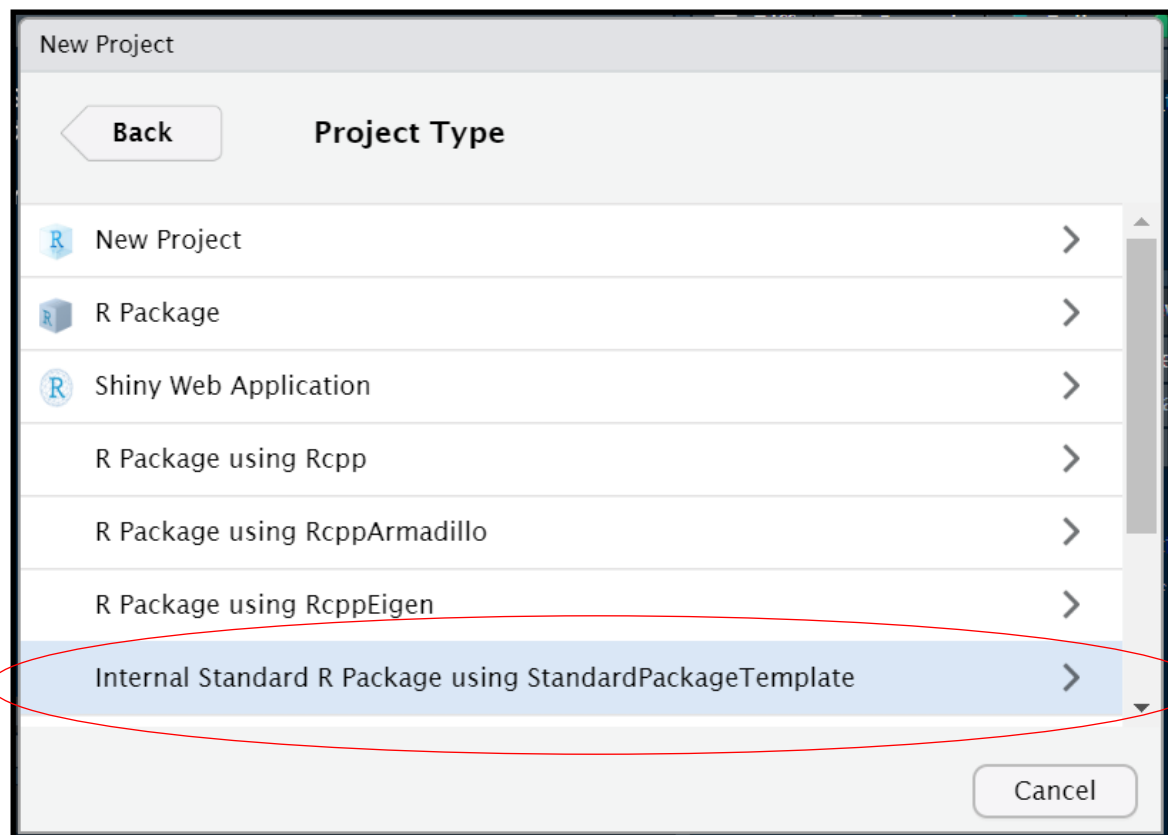
Retrieve code coverage percentage (to be updated in baamr_codecov_badge function):

- `x <- covr::package_coverage()`
- `covr::percent_coverage(x, by="file")`

Standard Package Folder Structure

Package Information	DESCRIPTION, README.md files
R function (with specification)	\R folder
Documentation	\man folder
Testing cases	\test folder
Examples	\vignettes folder
Change log	NEWS.md file

R-Studio Project Template Wizard



R-SDLC Cycle: Development Stage (1/2)

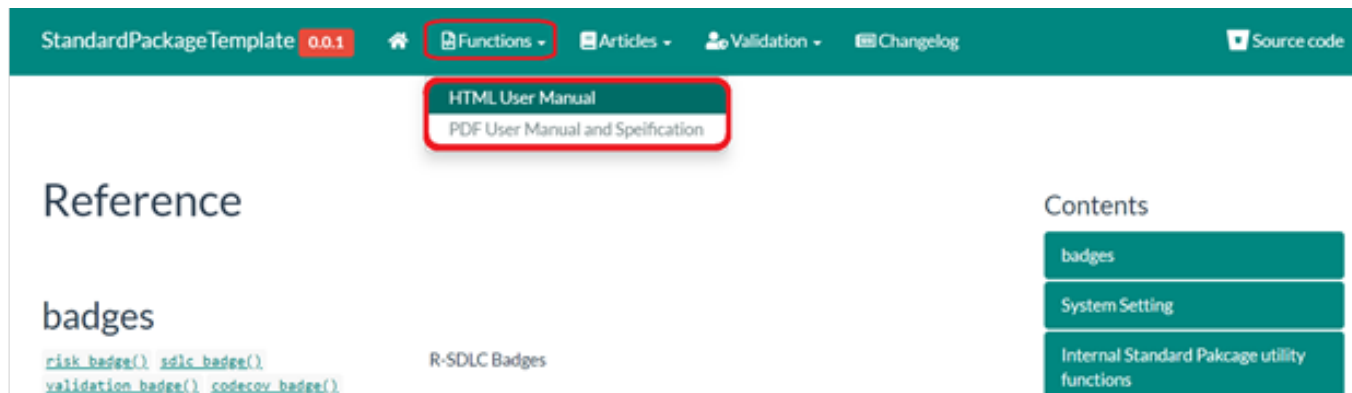
- Develop and follow consistent programming practice and design guidance
 - Example: Tidyverse style guide or design guide
- Define strategy for testing of R code/function
- Use of Standard package template to:
 - Plan and track the issue and errors fixed
 - Trigger automated builds and tests features
 - Track history when developer submits new code to a version controlled server (e.g. BitBucket)
 - An automation server (e.g. Jenkins) builds and tests the code
 - Generate updated badges for build status and code coverage based on the outcomes

The screenshot displays the StandardPackageTemplate web application. The top navigation bar includes links for Functions, Articles, Validation (highlighted with a red box and a dropdown menu), and Changelog. The dropdown menu for Validation shows options: Program Development and Validation Tracker, Plan and Track (selected), and Tests & Coverage Report. The main content area is titled 'Program Plan and Track' and features an 'Overall Program Status' section with filters for Column visibility and Excel. A search bar is also present. Below the search bar is a table with columns: Overall Program Status, Program Name, Round Number, and Description. The table shows a single entry: 'C' for Overall Program Status, 'basic_addition' for Program Name, '1' for Round Number, and 'Doing basic addition' for Description. On the right side, there is a 'Contents' sidebar with buttons for Overall Program Status, Define -> Develop, and Validate -> Operations.

Overall Program Status	Program Name	Round Number	Description
C	basic_addition	1	Doing basic addition

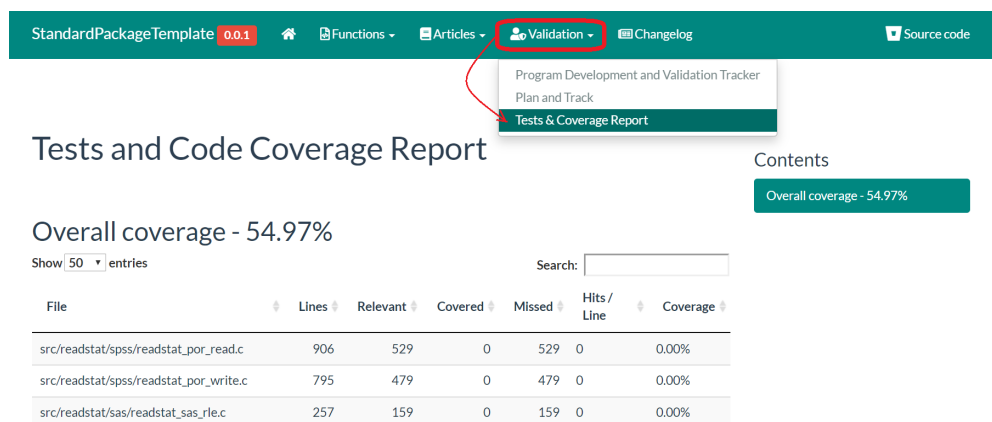
R-SDLC Cycle: Development Stage (2/2)

- New or updated R function or package should pass (display Pass badge) R package check list conditions with more than 80% (recommended) code coverage
- General recommendation is to develop a user manual for each function unless it is an internal function within an R package and or as determined by SME
 - User manuals illustrating a function's input, output and examples can be documented with the “roxygen2” package
 - Final user manual is automatically generated using roxygen2 and saved under Man Folder



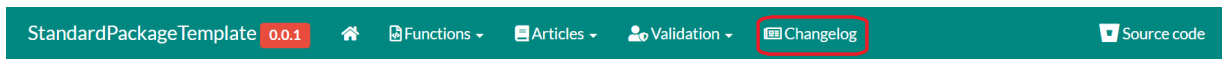
R-SDLC Cycle: Validation Stage

- Validation stage ensures the accuracy and integrity of developed or updated R functions in the regulated environment and makes the R package ready for production
- Based on validation plan the validator perform independent programming or double programming of key or all R functions
- Validator creates a new working branch in testthat folder using test-independent-testing-<filename>.R or test-double-programming-<filename>.R that matches the <filename> filename in the R folder to verify the results
- Validator conducts program code review for effectiveness and efficient programming practices to ensure best programming practices
- New or updated functions should pass with code coverage greater than 80% (recommended)
 - Create code coverage report using “covr” package to be displayed for user on R package webpage.



R-SDLC Cycle: Operation Stage

- Ensures the validated R package is installed properly into the computing environment by the R administrator
- R Administrator follows organization software release SOP
- Update news.md file for any changes to the function with unique version number. This is required to create changelog which will be posted on package website along with other information on the package
- If an R function is deprecated (only function or argument level applied) or retired, SME provide information on R function/argument name, date R function/argument will be deprecated or retired, reason for deprecating or retiring, workaround or alternate R function/argument



Changelog

Contents

0.0.1

StandardPackageTemplate 0.0.1

27-SEP-2019

- Added a `NEWS.md` file to track changes to the package.
- Initial Version

Package Website

- An integrated website to summarize user manual, SDLC documentations:
 - **Badge system**
 - **Function user manual**
 - **Reference articles**
 - **Program Development Validation Tracker/Plan and Track tables**
 - **Code coverage report**
 - **Change log**
 - **Source code**
- Function `build_site()` in package “pkgdown” can help build an Organization-theme website
- Function `deployApp()` in package “rsconnect” can deploly the website on an internal RStudio Connect Server

SDLC of Internal Standard R Package

	R package
Specification	Bitbucket to host and develop specification package
Development	Bitbucket hosts development version
Testing and Validation	Jenkins: cloud server to automatically run test case and validation code
Operation	RStudio Package Manager or Artifactory to host in production version
Delivery	Install compiled R package from RStudio Package Manager or Artifactory to a server host computing platform and read only to user.
Retire	Bitbucket to archive retired R internal packages in development. RStudio Package Manager or Artifactory to archive R package versions in production

Testing and Traceability (using ggplot2 as example)

ggplot2 part of the tidyverse 3.2.0

Overview

build passing build passing coverage 80% CRAN 3.2.0

ggplot2 is a system for declaratively creating graphics, based on [The Grammar of Graphics](#). You provide the data, and ggplot2 takes care of the details.

Reference Articles News Extensions

Aesthetic specifications

Extending ggplot2

Load from CRAN at <https://cloud.r-project.org/package=ggplot2>

<https://github.com/tidyverse/ggplot2>

History for [ggplot2](#) / R / aes.r

Commits on Jun 4, 2019

fix tidy styling in extract_target_is_likely_data(), improve parameter...
paleolimbot committed 24 days ago ✓

implement changes from review: better organized tests, improved argum...
paleolimbot committed 24 days ago ✓

<https://github.com/tidyverse/ggplot2/commits/master/R/aes.r>

Files	Coverage
aes-calculated.r	88.88%
aes.r	79.31%
annotation-custom.r	87.50%

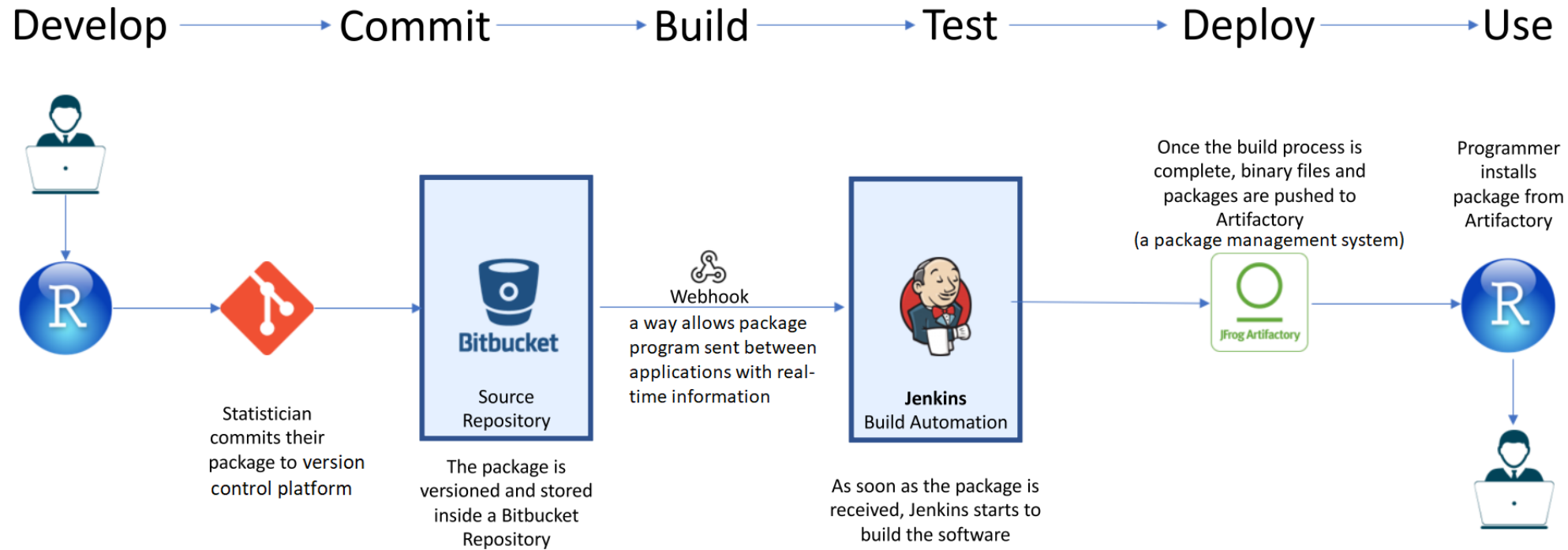
<https://codecov.io/gh/tidyverse/ggplot2/tree/master/R>

```
105 5 stopifnot(is.list(x))
106 5 x <- lapply(x, new_aesthetic, env = env)
107 5 structure(x, class = "uneval")
108 }
109
110 #' @export
111 print.uneval <- function(x, ...) {
112   cat("Aesthetic mapping: \n")
113
114   if (length(x) == 0) {
115     cat("<empty>\n")
116   } else {
```

<https://codecov.io/gh/tidyverse/ggplot2/src/master/R/aes.r>

- One website to cover all documentations
- All the testing rerun while code changes
- Full traceable development history
- Line by line report of the testing coverage

R Continuous Development and Integration (CI/CD)



Summary

- We propose to save all required documents in an R package
 - Specification, validation tracker, user manual etc.
- Badges system will be used to classify properties of an R package or function
- CI/CD technique simplified and automate version control, testing and traceability
- An integrated website is developed with features like badge system, user manual, vignettes, validation tracker, code coverage report, changelog, etc.

Package	
DESCRIPTION	Setup
R/	Write code
tests/	Test
man/	Document
vignettes/	Teach
data/	Add data
NAMESPACE	Organize

Backup

R SDLC Badge System (DONOT INCLUDE)

R-SDLC Badge	Definition
Specification	Define a package and function's scope, business requirements, developing environment, etc.
Development	Develop functions following programming style and best practice, complete developer testing and build user manual.
Validation	Validate a developing function.
Independent Testing	Create testing programs to run the function and verify the results.
Double Programming	Re-create function and testing programs to reproduce all output from the function using the specs only, without looking at the function being validated.
Customer Review	Execute new functions and review output, usually acted by statistician and/or business end users.
Stable	Indicate an R package/function is ready to release in production at current version.
Deprecated	Indicate an R package/function will be retired and may be replaced by a new R package/function in near future.
Retired	Indicate an R package is no longer used.
Code Coverage	Measure source code test/validation completion degree.
Build Status	Indicate whether the R package and corresponding functions pass build and check criteria or not.