

Applications of the R Package Validation Framework and {valtools}

ELLIS HUGHES

THE R PACKAGE VALIDATION FRAMEWORK WORKING GROUP

DVOST

What is validation?

Good Clinical Practices

ICH E6 - Section 5 – Sponsor

Ensure and document that the electronic data processing system(s) conforms to **the sponsor's established requirements** for completeness, accuracy, reliability, and consistent intended performance (i.e., validation)... (ICH E6 Section 5.0)

their approach to validation...takes into consideration the **intended use of the system** and the **potential of the system to affect human subject protection and reliability** of trial results. (ICH E6 Section 5.5.3)

Statistical Principles for Clinical Trials

ICH E9 – Section V – Data Analysis Considerations

The **credibility of the numerical results** of the analysis depends on the **quality and validity of the methods and software (both internally and externally written)** used both for data management (data entry, storage, verification, correction, and retrieval) and for processing the data statistically (ICH E9 Section 5.8 H)

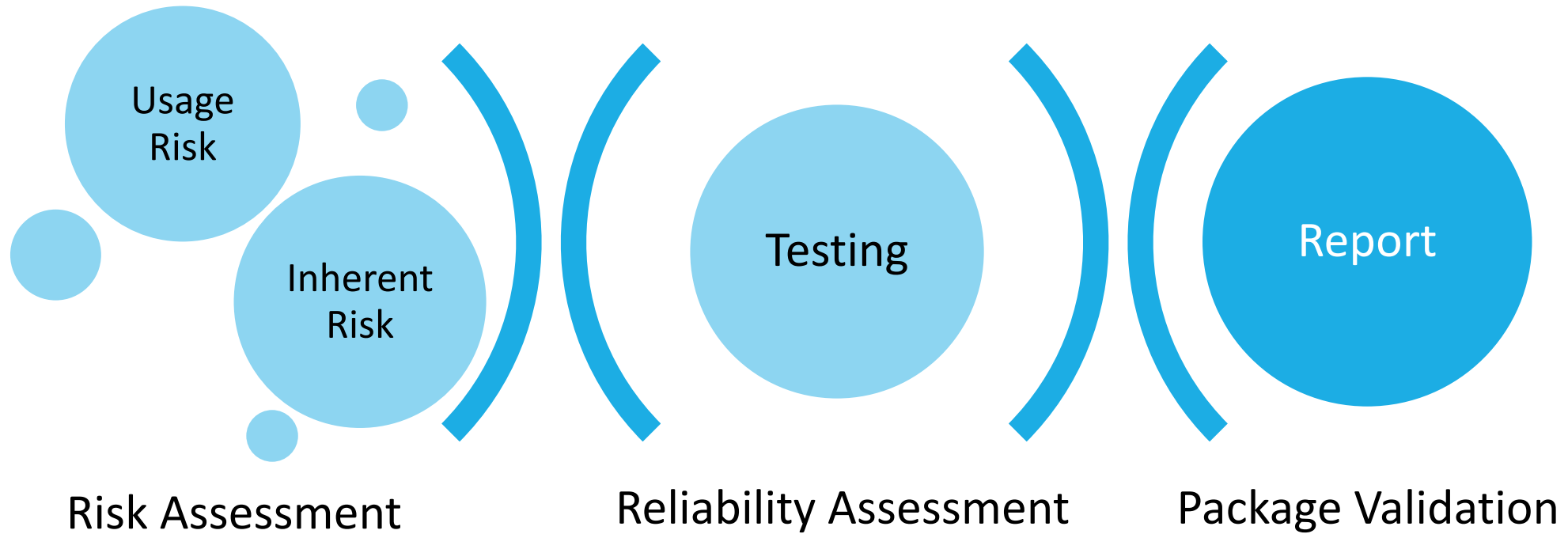
The computer software used for data management and statistical analysis should be reliable, and documentation of **appropriate software testing procedures should be available**. (ICH E9 Section 5.8 H)

Validation

- Definition: A process of establishing and documenting that the specified requirements of a computerized system can be consistently fulfilled... (ICH E6 1.65)
- Risk Based Approach to Validation
 - ICH E6 Section 5.0
 - ICH E6 Section 5.5.3
- Statistical analysis should be reliable
 - ICH E9 Section 5.8 H

Package validation?

Package Validation Considerations



Risk Assessment – Usage Risk

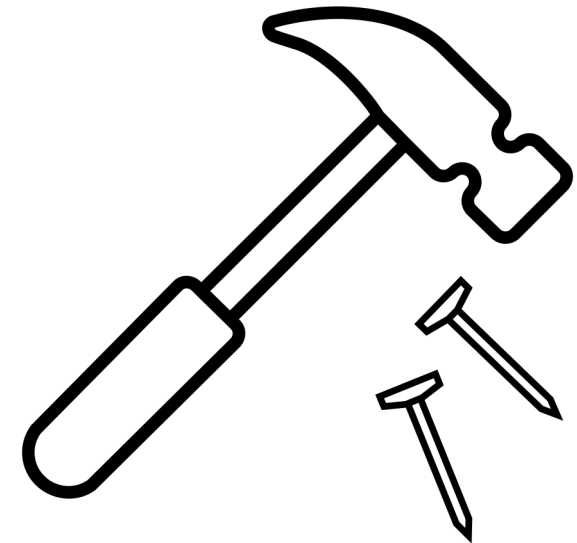
Intended use drives how much inspection is required

Requirements and usage goals

Byproducts

Where used

Describes the usage risk of the package, defined by the organization



Risk Assessment – Inherent Risk

What is the quality and trust in the package

Measurables

- Test coverage, documentation existing, vignettes

Perception

- Authors, download trends, community response

{riskmetric} & R Validation Hub

Describe the inherent risk of the package, generic risk can be defined generally



Reliability Assessment – Testing

Proof requirements are met

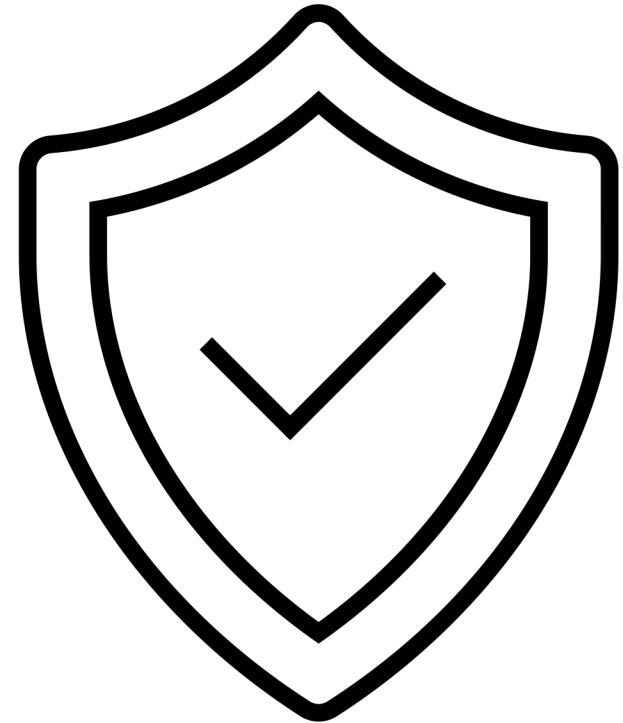
Tests based on package risk (mitigation)

Specific to risks identified

Reasonable use cases

Quantifiable & reviewable

- Known outputs from trustworthy sources
- Cross reference against other software



Report – Package Validation

Documentation of Reliability

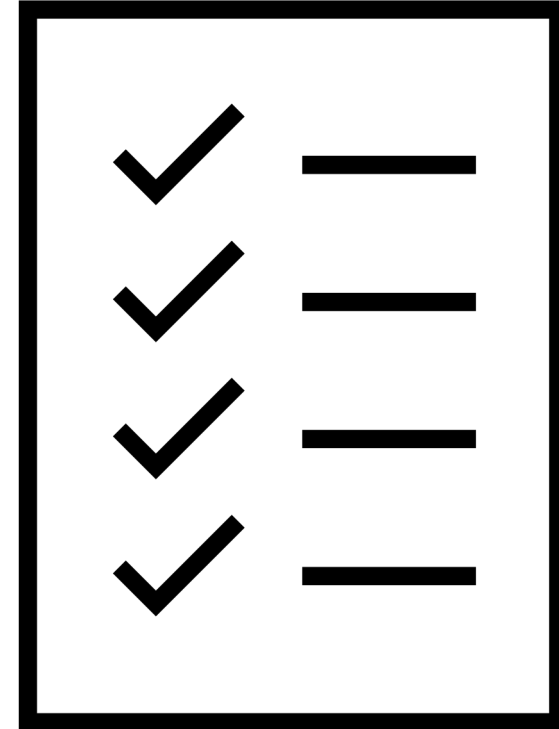
Environment

Reviewer

Risk assessment

Testing results

- Tests pass (%)
- Repeatable and reproducible



R Package Validation Framework

Organized, consistent format

Self contained

Flexible to organizations needs

Reproducible

White Paper

- https://bit.ly/RPackage_Validation



R Package Validation Framework

Internal Packages



Validation concerns for R Packages
used in clinical trials are consistent for
internally and **externally** developed
packages, **and across pharma**

R Package Validation Framework

External Packages

Focus on organization of contents

Requirements (As required)

Test Cases (As required)

Test Code (Based on risk assessed)

Package Assessment Document

- Risk Assessment
- Results of test code
- Reproducible

```
· | - validation
· | | - requirements/
· | | - test_cases/
· | | - test_code/
· | | - validation.yml
· | | - change_log.md
· | | - .gitignore
· |
· | - validation.Rmd
```

R Package Validation Framework

External Packages



{valtools} Demo Day

{cowsay} - Package Assessment

```
> cowsay::say("Hello PHUSE US Connect 2022!", by = "yoda")
```

```
-----  
Hello PHUSE US Connect 2022!
```

{cowsay} - Package Assessment - Getting Started

```
> valtools::vt_create_packet(path = "PHUSE-valtools-demo",target = "cowsay")
v Setting active project to '~/PHUSE-valtools-demo'
v Writing 'PHUSE-valtools-demo.Rproj'
v Adding '.Rproj.user' to '.gitignore'
v Setting active project to '<no active project>'
v Adding '!*' to 'PHUSE-valtools-demo/./validation/.gitignore'
v Setting active project to '<no active project>'
Created ./validation in package structure
validation config file created. Add user information through `vt_add_user_to_config()`
Created validation packet
v Opening 'PHUSE-valtools-demo/' in new RStudio session
```

{cowsay} - Package Assessment - Report

Console

Terminal x

Jobs x

R 4.1.1 · ~/PHUSE-valtools-demo/ ↩

```
> valtools::vt_use_report(template = "packet")
```

```
* Modify './validation.Rmd'
```

```
> |
```

validation.Rmd x

← → ↶ ↷ ABC 🔍 Knit ⚙

+c

↑

↓

Run

↺

≡

A

```
1 |---  
2 title: Validation Report for cowsay  
3 author: Ellis Hughes  
4 date: "`r Sys.Date()`"  
5 output:
```

{cowsay} - Package Assessment - Report

```
84
85 ▾ ## Package Information
86
87 ▾ ### Change Log
88
89 ▾ ``{r change-log-table}
90 vt_scrape_change_log() %>%
91   vt_kable_change_log()
92 ▲ ``
93
94 ▾ ### Validation Environment
95
96 ▾ ``{r env-table}
97 vt_scrape_val_env() %>%
98   vt_kable_val_env()
99 ▲ ``
100
101
```

{cowsay} - Risk Assessment – Inherent Risk

Prints character strings with ASCII art of “animals”

Available on CRAN since 2015-10-13

Reliable 84000 downloads total

Documentation of all exported functions

Author list is qualified (Scott Chamberlain, et al)

Test coverage of 62%



{cowsay} - Risk Assessment – Usage Risk

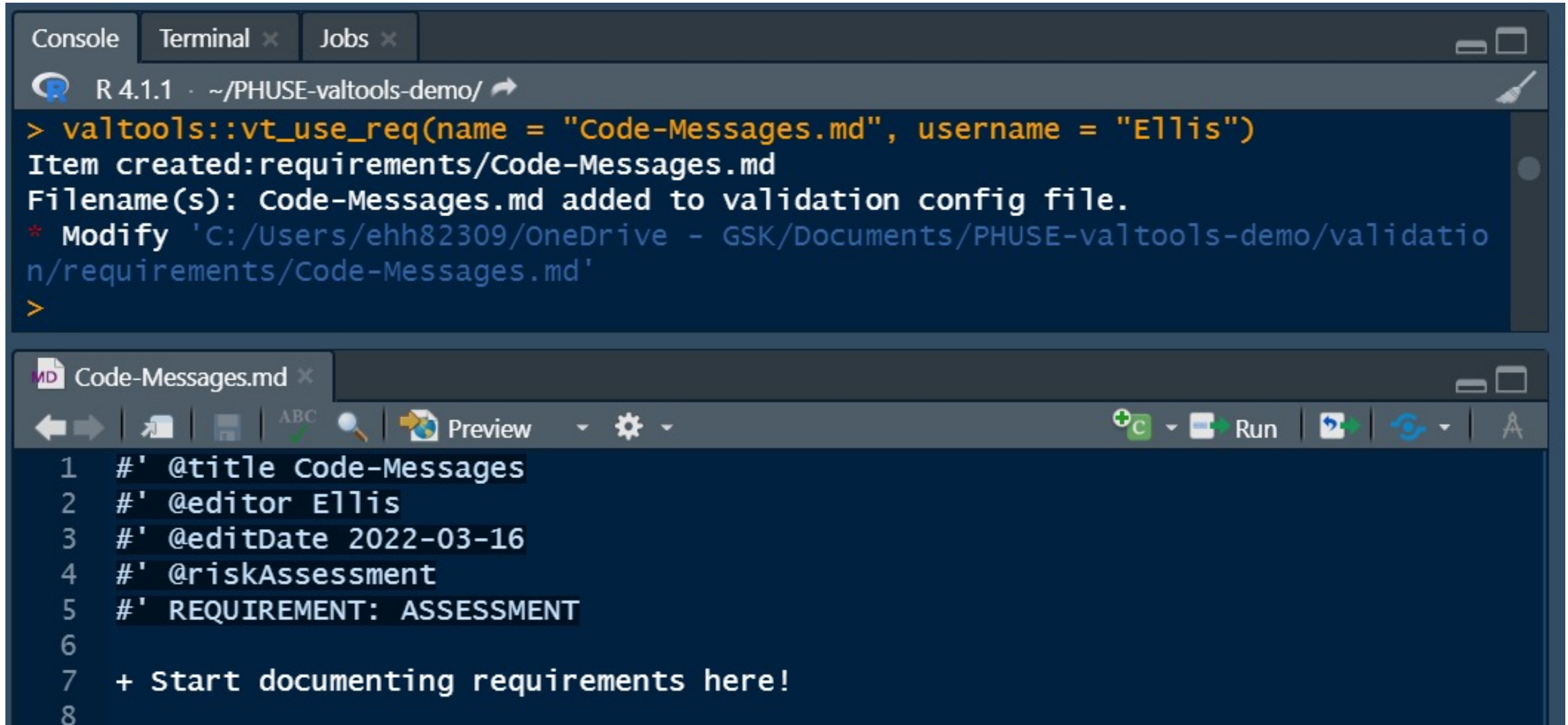
{cowsay} used to provide “messages with flair” in code logs

Unlikely/impossible to impact any findings (main, support, research)

```
> cowsay::say("Hi Howdy, I;m the {cowsay} package!", by = "cow")

-----
Hi Howdy, I;m the {cowsay} package!
-----
  \      ^__^
   \      (oo)\_____
      (____)\       )\/\
              ||----w |
              ||     ||
```

{cowsay} - Package Assessment

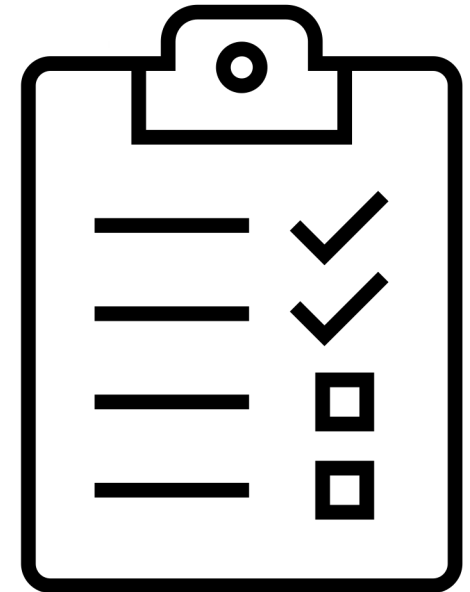


The image shows a screenshot of an R console window and a markdown editor window. The R console window is titled "Console" and shows the execution of the `valtools::vt_use_req` function. The output indicates that a file named `Code-Messages.md` has been created and added to the validation configuration file. The markdown editor window is titled "Code-Messages.md" and shows the content of the file, which is a validation requirement for the package `{cowsay}`.

```
> valtools::vt_use_req(name = "Code-Messages.md", username = "Ellis")
Item created:requirements/Code-Messages.md
Filename(s): Code-Messages.md added to validation config file.
* Modify 'C:/Users/ehh82309/OneDrive - GSK/Documents/PHUSE-valtools-demo/validation/requirements/Code-Messages.md'
>
```

```
1 #' @title Code-Messages
2 #' @editor Ellis
3 #' @editDate 2022-03-16
4 #' @riskAssessment
5 #' REQUIREMENT: ASSESSMENT
6
7 + Start documenting requirements here!
8
```

{cowsay} has a **relatively low inherent risk** due to highly credentialed developers, quality documentation and vignettes, lack of impact on findings, and a rather silly use case



{cowsay} - Reliability Assessment

Because of low test coverage, may want to add additional tests

Tests serve to

- Confirm behavior on system
- Address potential risks

Depending on organization

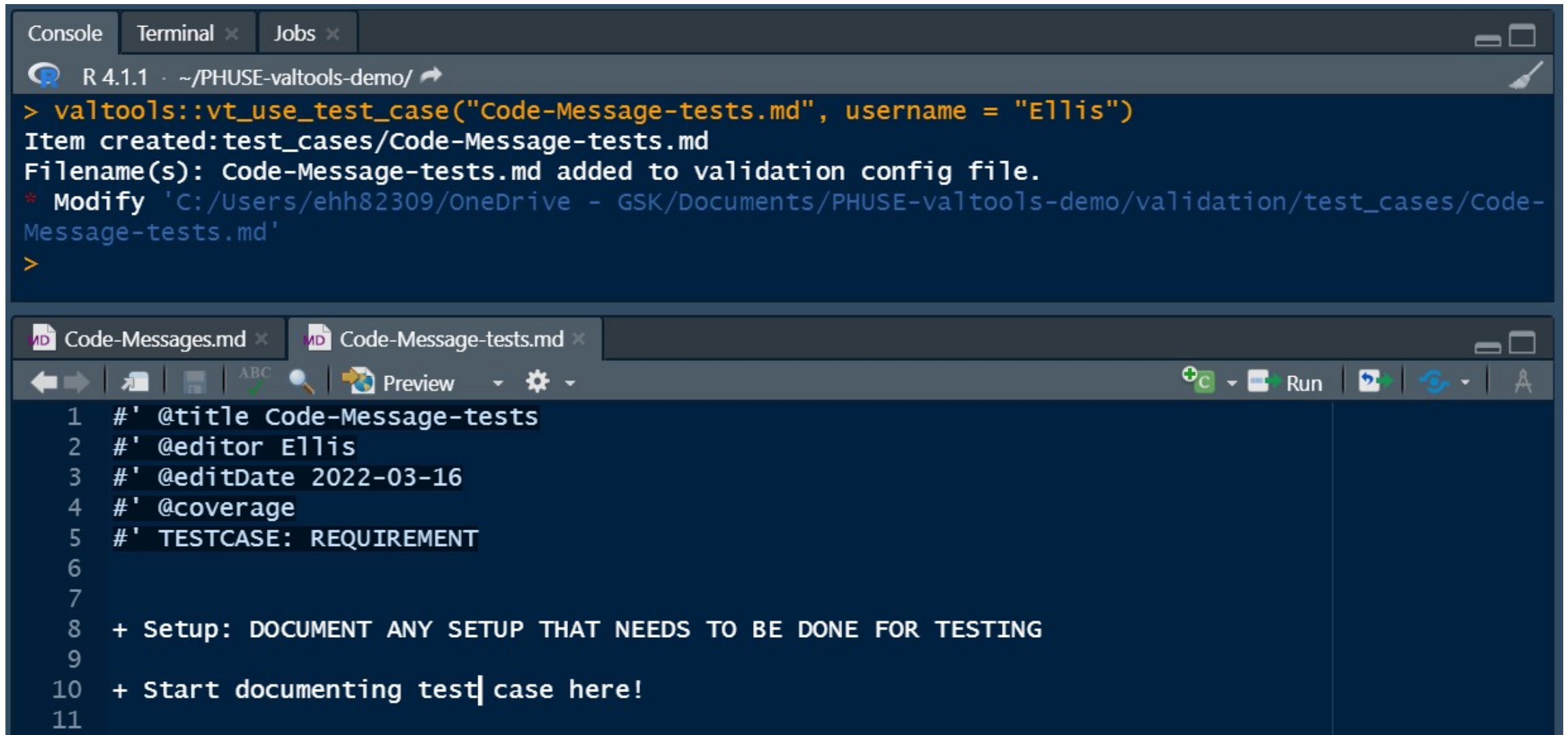
- Testing plan may be required
- Tests may need to be reviewed by another qualified programmer

```
> cowsay::say("{cowsay} is to be trusted", by = "hypnotoad")
```

```
-----  
{cowsay} is to be trusted  
-----
```



{cowsay} - Reliability Assessment - Testing Plan



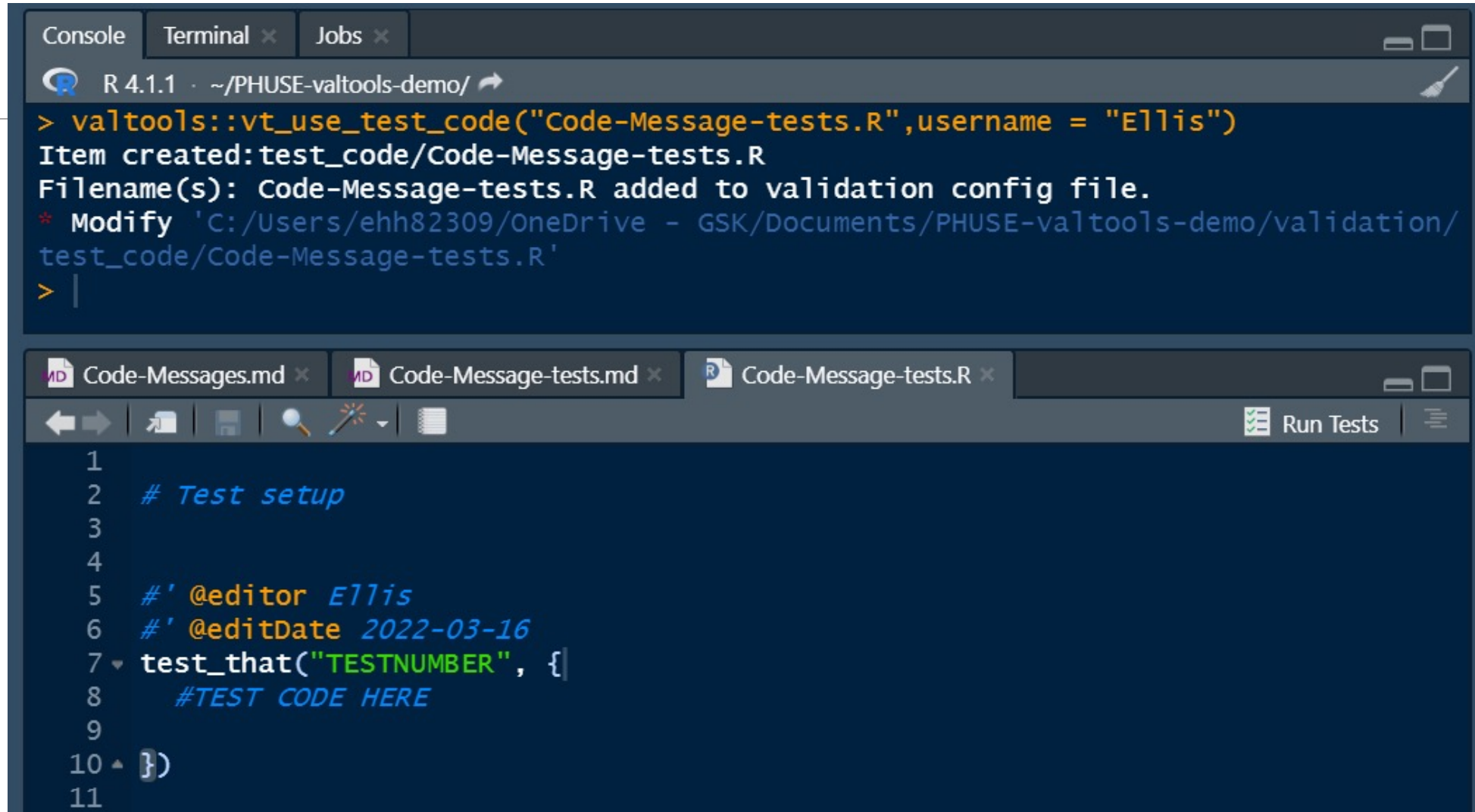
The image shows a screenshot of an R console window and a markdown editor. The console window at the top shows the execution of the `valtools::vt_use_test_case` function, which creates a test case file named `Code-Message-tests.md` and adds its filename to the validation configuration file. The editor window below shows the content of the created file, which includes metadata like title, editor, edit date, and coverage, followed by instructions for documenting the test case.

```
R 4.1.1 · ~/PHUSE-valtools-demo/
> valtools::vt_use_test_case("Code-Message-tests.md", username = "Ellis")
Item created:test_cases/Code-Message-tests.md
Filename(s): Code-Message-tests.md added to validation config file.
* Modify 'C:/Users/ehh82309/OneDrive - GSK/Documents/PHUSE-valtools-demo/validation/test_cases/Code-Message-tests.md'
>
```

```
Code-Messages.md x Code-Message-tests.md x
← → | 📄 | 📁 | ABC | 🔍 | 🌐 Preview | ⚙️ |
+ C - Run | ↺ | 🔁 | 🎯

1 #' @title Code-Message-tests
2 #' @editor Ellis
3 #' @editDate 2022-03-16
4 #' @coverage
5 #' TESTCASE: REQUIREMENT
6
7
8 + Setup: DOCUMENT ANY SETUP THAT NEEDS TO BE DONE FOR TESTING
9
10 + Start documenting test case here!
11
```

{cowsay} - Reliability Assessment - Tests



```
R 4.1.1 · ~/PHUSE-valtools-demo/

> valtools::vt_use_test_code("Code-Message-tests.R",username = "Ellis")
Item created:test_code/Code-Message-tests.R
Filename(s): Code-Message-tests.R added to validation config file.
* Modify 'C:/Users/ehh82309/OneDrive - GSK/Documents/PHUSE-valtools-demo/validation/
test_code/Code-Message-tests.R'
> |
```

```
Code-Messages.md × Code-Message-tests.md × Code-Message-tests.R ×
Run Tests

1
2 # Test setup
3
4
5 #' @editor Ellis
6 #' @editDate 2022-03-16
7 test_that("TESTNUMBER", {
8   #TEST CODE HERE
9
10 })
11
```

{cowsay} - Reliability Assessment – Tests Example

```
Code-Message-tests.R x
1 #' @editor E77is
2 #' @editDate 2022-03-16
3 test_that("message type outputs",{
4
5   expect_message(
6     cowsay::say("This is a test of the automated messaging system", type = "message"),
7     paste0(
8       c(
9         "",
10        "-----",
11        "This is a test of the automated messaging system ",
12        "-----",
13        "  \\",
14        "   \\",
15        "    \\",
16        "     \\\_\\_/|",
17        "    ==) ^Y^ (=)",
18        "     \\\  ^  /",
19        "      )=*(",
20        "      /  \\",
21        "      |  |",
22        "     /| | | \\\\",
23        "    \\\| | | | \\\\",
24        "   jgs //_// _/",
25        "      \\\_)",
26        ""
27      ),
28      collapse = "\n"
29    ),
30    fixed = TRUE # make checking for exact string, not regex
31  )
32
33 })
34
35
19:30 (Top Level) R Script
```

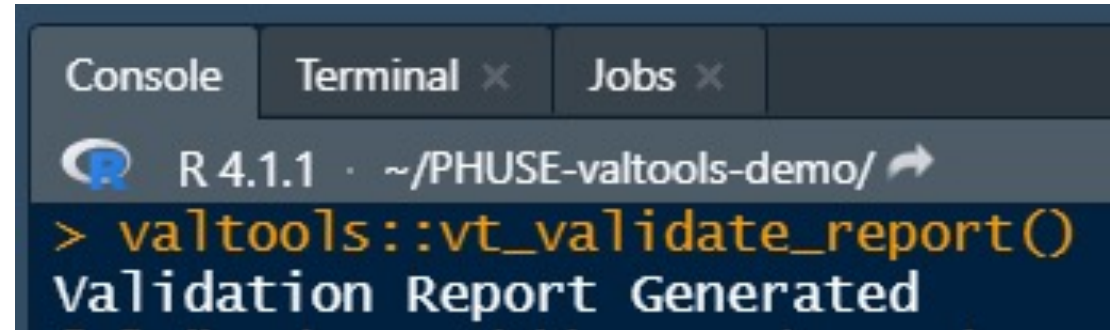
{cowsay} - Package Assessment – Validation

Reproducibly generate report

Combines assessment elements together

Executes tests in clean R environment

Automatically tags report for target and date



```
Console Terminal x Jobs x
R 4.1.1 · ~/PHUSE-valtools-demo/ ↵
> valtools::vt_validate_report()
Validation Report Generated
```

Validation Report for cowsay

Ellis Hughes

2022-04-19

Contents

1	Certifications	1
1.1	Signatures	1
2	Release details	2
2.1	Package Information	2
3	Assessment	3
4	Validation	3

1 Certifications

1.1 Signatures

AUTHOR My signature designates authorship of this document.

Role	Name and Title	Signature	Date
Validation Lead	Ellis Hughes, Lead		

2 Release details

2.1 Package Information

2.1.1 Validation Environment

Type	Resource	Version Detail
system	OS	Windows 10 x64 (build 19042)
	R	4.1.1
package_req	crayon	1.4.1
	fortunes	1.5.4
	rmsfact	0.0.3
extended_req	curl	4.3.2
	jsonlite	1.7.2
	knitr	1.36
	multicolor	Not Installed
	rmarkdown	2.11
	testthat	3.0.4
session	cowsay	0.8.0
	devtools	2.4.2
	kableExtra	1.3.4
	magrittr	2.0.1
	usethis	2.1.5
	valtools	0.4.0

2.2 Authors

2.2.1 Requirements

Requirement ID	Editor	Edit Date
Code-Messages	Ellis	2022-03-16

3 Assessment

- Prints character strings with ASCII art of “animals”
- Available on CRAN since 2015-10-13
- Reliable 84000 downloads total
- Documentation of all exported functions
- Author list is qualified (Scott Chamberlain, et al)
- Test coverage of 62%

4 Validation

- Code written at our organization must provide humorous messages
- Setup: DOCUMENT ANY SETUP THAT NEEDS TO BE DONE FOR TESTING
- Start documenting test case here!

Test	Results	Pass/Fail
message type outputs.1	As expected	Pass

{valtools} Resources

Website: <https://phuse-org.github.io/valtools/>

- Getting started vignette with more details and example

Workshop: https://bit.ly/valtools_workshop

- Focus on external package validation (discussed here)
- Slides and example problems with solutions

Source Code: <https://github.com/phuse-org/valtools>