

Good Programming Practices, Design and Agile in the Era of AI-Generated Code

UseR! 2026

Maciej Nasinski, PhD

UCB Poland



Context

1. What AI changed

AI made it easier to get **working code**.

2. What did not change

To get the **right code**, developers and AI still need domain knowledge, delivery discipline, and stakeholder alignment.

3. Core risk

The hardest gap remains the distance between a **prototype** or **MVP** and a **production-quality solution**.

Building Software with Proper AI Setup

AI setup matters

Strong AI outcomes require strong setup: **instructions, context, tools and MCP**. Setup should include **stakeholder needs** and **domain and technical knowledge**.

Core professional skill in AI-era R

Spec-driven development is the core professional skill in AI-era R.

```
a-package/                                # repo root
  .github/
    copilot-instructions.md  # persistent context
    instructions/
    prompts/
    agents/
  ...
```

Agile Scrum and Delivery Process

Story

Agile Scrum is designed for **complex products** and still requires **human interaction**. In this context, AI is valuable when used properly **with a team**.

Switch to human-led team delivery when

- Stakeholder needs are complex or evolving
- External users, critical outcomes, or compliance constraints
- Long-term ownership, auditability, and trust are required



Source: Gemini

Delivery reality

AI can accelerate drafts, but architecture, risk ownership, and stakeholder alignment remain human responsibilities.

Case Study: The gridify Package

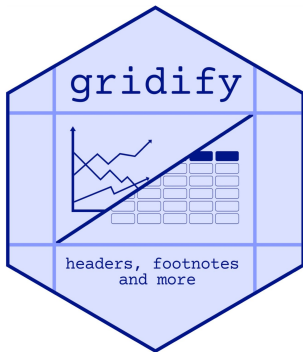
R-CMD-check.yaml passing

codecov 100%

pharmaverse gridify

cran v0.8.1

downloads 1051/month



Source: Own work

What gridify does

gridify builds on base R grid and adds flexible, customizable information around figures and tables using predefined or custom layouts. **Technical proof:** S4 classes + composition over inheritance + minimal dependency footprint.

Collaboration

Built through team collaboration and iterative delivery. Thanks to all gridify contributors, including those listed in the DESCRIPTION file and the README Acknowledgements section.

How we build gridify: from AI prototype to production-ready package

Prototype build sequence

1. **Align:** requirements and stakeholder expectations first
2. **Design:** UML + specifications + UX
3. **Challenge the design:** use **AI** to challenge and refine the design
4. **Prototype:** **AI**-assisted draft based on design and requirements
5. **Refine:** collaborative review and improvements

Next steps to production

Review, inspect, adapt

- **Stakeholder engagement**
- **Small, reviewable PRs**
- **Proper AI setup in .github**
- **Transparent AI PR tags**
- **At least two approvals per PR**
- **Advanced CI checks** (e.g. test coverage, code quality)

What we got from the process: Successful gridify delivery

downloads 1051/month



pharmaverse gridify

PharmaCorp International
ONCOLOGY / Advanced NSCLC
Study PC-2025-001

CONFIDENTIAL

Final
Data Cut-off: 2025-09-30

Table 14.1.1
Summary of Demographics and Baseline Characteristics
Baseline Biomarker 1 (BMRKR1) by Geographic Region
Safety Analysis Set

	Region	n	BMRKR1
A: Drug X	Africa	8	6.47 (4.70)
	Asia	91	6.00 (3.32)
	Eurasia	5	4.88 (1.61)
	Europe	4	5.44 (4.48)
	North America	13	6.84 (4.92)
	South America	13	5.18 (3.51)
B: Placebo	Africa	7	4.65 (1.92)
	Asia	94	5.97 (3.53)
	Eurasia	8	6.35 (3.89)
	Europe	3	4.17 (0.47)
	North America	15	4.21 (1.62)
	South America	7	6.22 (3.44)
C: Combination	Africa	11	6.57 (3.24)
	Asia	83	5.44 (3.53)
	Eurasia	6	3.58 (0.87)
	Europe	2	2.35 (0.42)
	North America	20	6.26 (4.13)
	South America	10	6.62 (2.71)

Note: BMRKR1 values shown as Mean (SD).

Source: ADSL dataset random.cdisc.data
Program: table_posit.R, 2025-10-02 21:50 CEST

Page 1 of 1

Stakeholder feedback

Positive feedback from stakeholders.

The Best Pharma Table 2025

Strong engineering practices produce strong communication artifacts.

Internal quality scales delivery

High-quality internals made delivery safer, maintenance easier, and extension faster.

Source: Own work

Example: Applying AI for a new gridify feature delivery

Initial AI proposal

- Concrete case: **gridify PR #18**
- Feature drafted with solid Copilot setup in .github and a detailed context and instructions
- The draft ready for review

What production required

- **Dozens of commits** after the first draft
- Updating for package architecture and consistency
- Review feedback and stakeholder-driven adjustments

Lesson  →  

AI created a starting draft (PR #18); **team iteration** turned it into a reliable production feature.

NOTE: AI-assisted PRs trigger more change requests on average.

Takeaways

AI alone can not be relied upon to produce the entire complex solution.

- **Design-first:** requirements and architecture before generation
- **Review-first delivery:** inspect, adapt, test, and iterate with stakeholders
- **Human ownership:** final quality, risk, and production decisions stay with developers
- **Developer as workflow architect:** owns the communication loop from stakeholder needs to resilient specification

Thank you



Scan for gridify