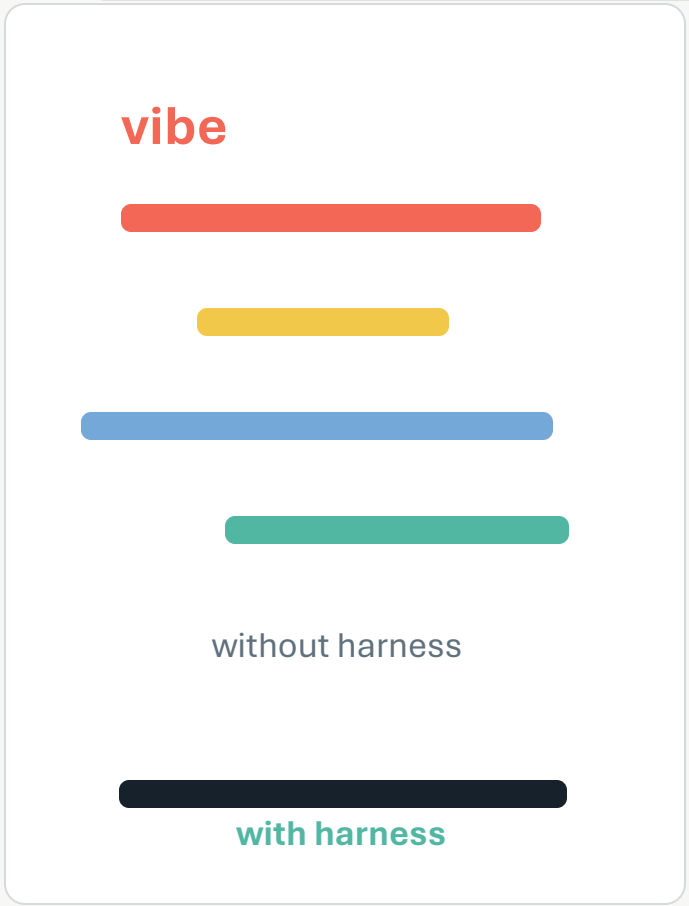
Implement Harness Engineering

A clear path to minimize rabbit holes while vibing



Vibing is not the problem.
Unmanaged vibing is.

15-20 min talk

non-technical builders

copy-ready template

THE PAIN

Rabbit holes happen when AI has motion without rails.

Goal drift

The task quietly changes from "simple app" to framework archaeology.

Missing context

AI guesses your domain, tools, terms, and risk tolerance.

No stop rule

The chat keeps expanding because nobody defined "done".

No review loop

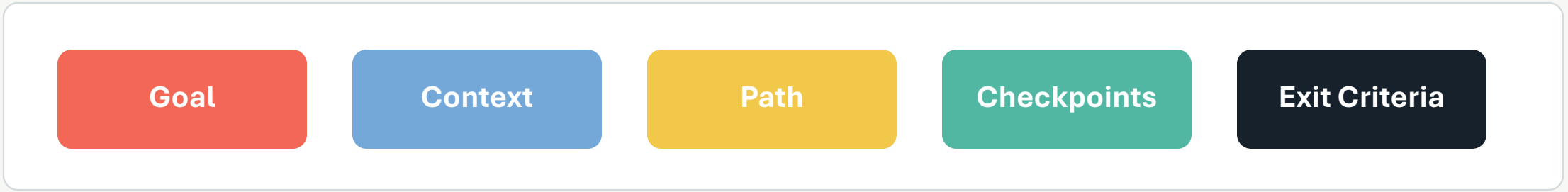
Changes pile up before you understand what changed.

"I asked AI to build an app, now I have 12 files, 6 errors, and no idea what happened."

THE REFRAME

Harness engineering turns vibe into repeatable workflow.

Harness = reusable constraints around AI work



Do not prompt harder. Harness better.

BEFORE / AFTER

Same intention. Different operating system.

Naked prompt

Build me an app for tracking patients / chemicals / animation assets.

Likely result: goal drift, tool guessing, too many files, unclear finish line.

Harnessed prompt

Act as my build guide. First ask 3 scoping questions. Then propose a 3-step path. After each step, summarize changes, remaining work, and risk.

Likely result: shared plan, smaller steps, visible progress, clear done point.

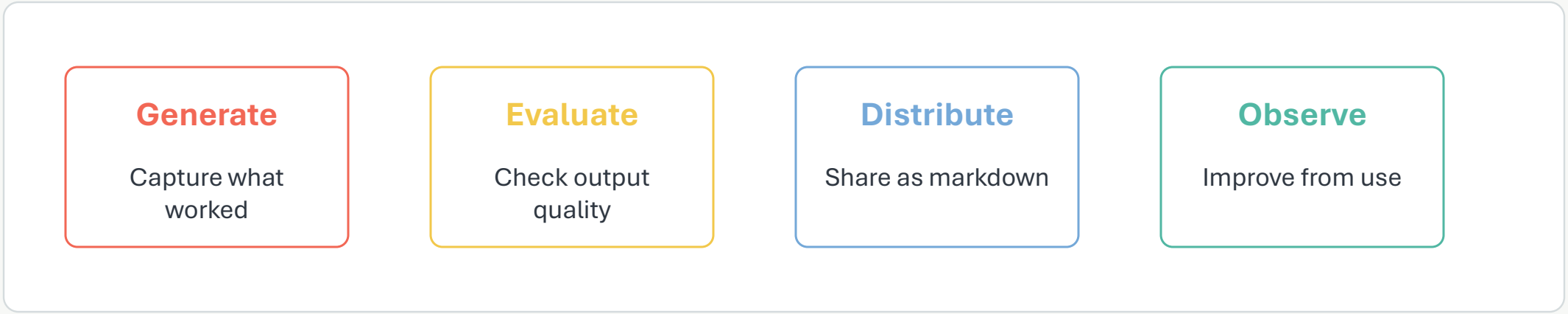
COPY THIS

A harness prompt is a small reusable contract.

GOAL	Build [specific outcome] for [specific user] by [time / quality bar].
CONTEXT	Use this domain background, terms, examples, and files: [paste/link].
CONSTRAINTS	Stay within [tool, budget, safety, style, compliance, time].
ASK FIRST	Before acting, ask only questions that change the plan.
CHECKPOINTS	After each major step: what changed, what remains, what risk.
DONE	Stop when [acceptance criteria] pass. Do not expand scope.

CONTEXT IS THE NEW CODE

The library is not just prompts. It is the operating loop.



Context becomes an asset when teams can reuse, review, and improve it.

OPEN STANDARD

A context pack can be plain markdown.

context-pack/

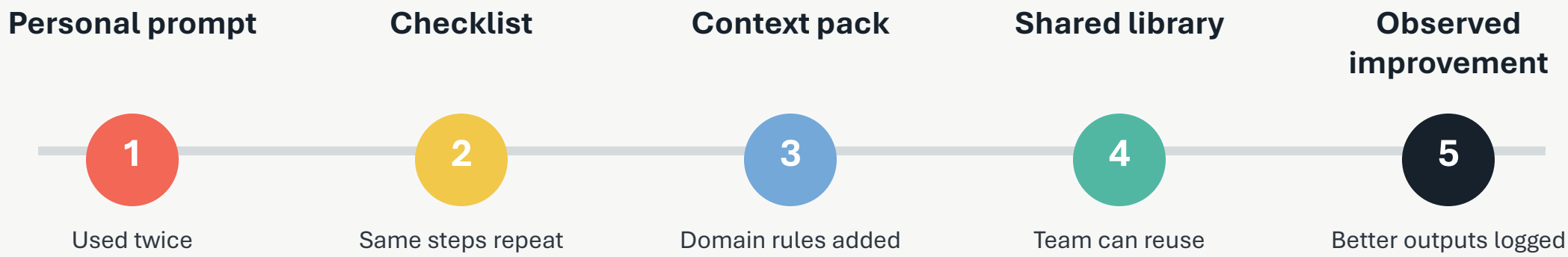
README.md	what this pack is for
DOMAIN.md	terms, rules, assumptions
PROMPTS.md	reusable prompts
EXAMPLES.md	good and bad outputs
CHECKPOINTS.md	review gates
GEMINI.md	language handoff notes

Portable across tools

No lock-in required.
Start with files you can read, edit, version, and share.

LIBRARY LADDER

Grow the library only when reuse proves it deserves to exist.



Do not start with a giant library. Start with one workflow you repeat.

ANTI-RABBIT-HOLE RULE

Make every major step leave a visible checkpoint.

After each major step, summarize:

- What changed?
- What remains?
- Where is risk?

This is cheap insurance: it turns hidden motion into visible progress.

NEXT STAGE

Turn one repeated task into one context pack.

Today

Pick one task you repeat often.

This week

Build one context pack. Use it three times.

Next week

Improve it once. Share it with one person.

Your expertise is the context. Harness engineering is how you make AI use it without dragging you into the hole.