

Context Inheritance

A Git-native architecture and pre-registered study of repository context for AI coding agents

Pavel Kerbel

Milana Kerbel · Vitali Abramov · Liat Abramov

INDEPENDENT RESEARCH IN AGENTIC SOFTWARE DEVELOPMENT



Excellent résumé. First day on the team.



GENERAL KNOWLEDGE

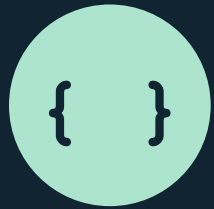
Knows the language.

REPOSITORY KNOWLEDGE

Doesn't know our why.

“Why did we reject this approach?”

And then the session ends.



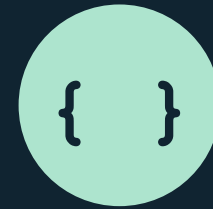
SESSION 01

Work completed.



THE HANDOVER

Nothing carried forward.



SESSION 02

Learn it again.

Unless we deliberately preserve what the session learned.

What I wanted from context.

More than memory: history, branching, blame, review, rollback.

History

Versioned snapshots

Branching

Blame / provenance

Human review

Rollback

Instruction files

AGENTS.md · CLAUDE.md

RAG / retrieval

Search and augmentation

Agent memory

Session-to-session continuity

I wasn't describing a better prompt file. I was describing source-control behavior.

I built a brain. Then I remembered Git.

ANOTHER SERVICE

Agent → MCP → Database

Operate. Integrate. Approve.



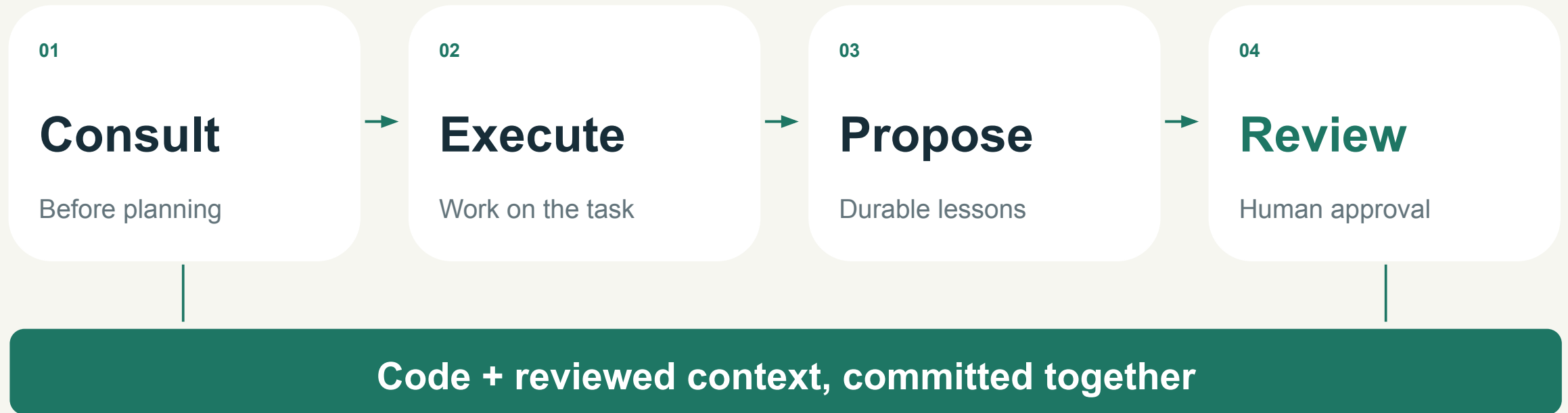
AN EXISTING WORKFLOW

Files → Git → Pull request

Version. Branch. Review.

MCP can remain a serving layer. The core does not require it.

The repository becomes the handover.



Three sections. One contract.

Intent

What are we preserving?

Constraints

What must not change—and why?

Evolved Context

What did we learn?

CONTEXT.MD · ILLUSTRATIVE

```
# Repository Context
```

```
## Intent
```

```
Preserve offline repairability.
```

```
## Constraints
```

```
No ORM: query opacity broke  
offline repair. Use raw SQL.
```

```
## Evolved Context
```

```
[2026-06-29] pkg x >=3 breaks  
ARM64; pin 2.3.x.
```

Does RCL actually help?

Separate experiments. Three questions.

01 / AUTHOR

Who writes it?

02 / DELIVERY

How is it read?

03 / READER + TASK

When does it help?

WITHIN EACH COMPARISON

Same task + same executor

No context

vs.

With context

WE MEASURED

Consultation · Localization · Repair success · Tokens

A real task: keep the attributes.

The values were selected. The metadata was lost.

MAINTAINER'S REGRESSION TEST · SIMPLIFIED

INPUT	VALUES	ATTRIBUTES
cond	[True, False]	{"attr": "cond"}
x	[1, 1]	{"attr": "x"}
y	[0, 0]	{"attr": "y"}

expected [1, 0] {"attr": "x"}

REQUESTED OPTION → MAINTAINER FIX

```
xr.where(cond, x, y,  
         keep_attrs=True)
```

Pass the policy to `apply_ufunc`.
Preserve `x`'s attributes—not `cond`'s.

`xarray/core/computation.py`

LOCALIZATION

**Patch touches a file changed
in the reference repair.**

RESOLVE

**Repair passes the official
benchmark evaluation.**

Same agent. Different starting point.

Gemma 4 executes both runs. Only the supplied guidance changes.

NO ADDITIONAL CONTEXT · B

No repository guide

```
grep -r "where"  
xarray/core/core.py
```

No patch

Later read computation.py too.

OPUS 4.8 GUIDANCE → GEMMA 4 · E

Prepared blind; injected into the prompt

```
grep -r "keep_attrs"  
xarray/core/computation.py
```

624-byte patch

First search targeted keep_attrs.

Both unresolved.

Selected illustration, not a successful-fix demonstration.

Who writes the context matters.

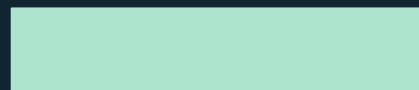
Gold-file localization

No context



21.1%

Opus-authored
blind seed



47.2%

+26.1 pp

123.70% increase ($p < 0.0001$)

SEPARATE COMPARISON

**Gemma's own
lessons**

+0.0 pp

**A file is useful
only if it is read.**

97%

GEMMA 4 · LOCAL DELIVERY

File and sharded arms

100%

OPUS 4.8 · BROAD SAMPLE

File and sharded arms

Consultation ≠ comprehension.

Selective reading: What we experienced

Context tokens per episode



EXPLORATORY PERFORMANCE

46.8% → 53.4%

Monolith → sharded
gold-file localization

**And sometimes
there is no room for improvement**

90.9%

Repair success without additional context

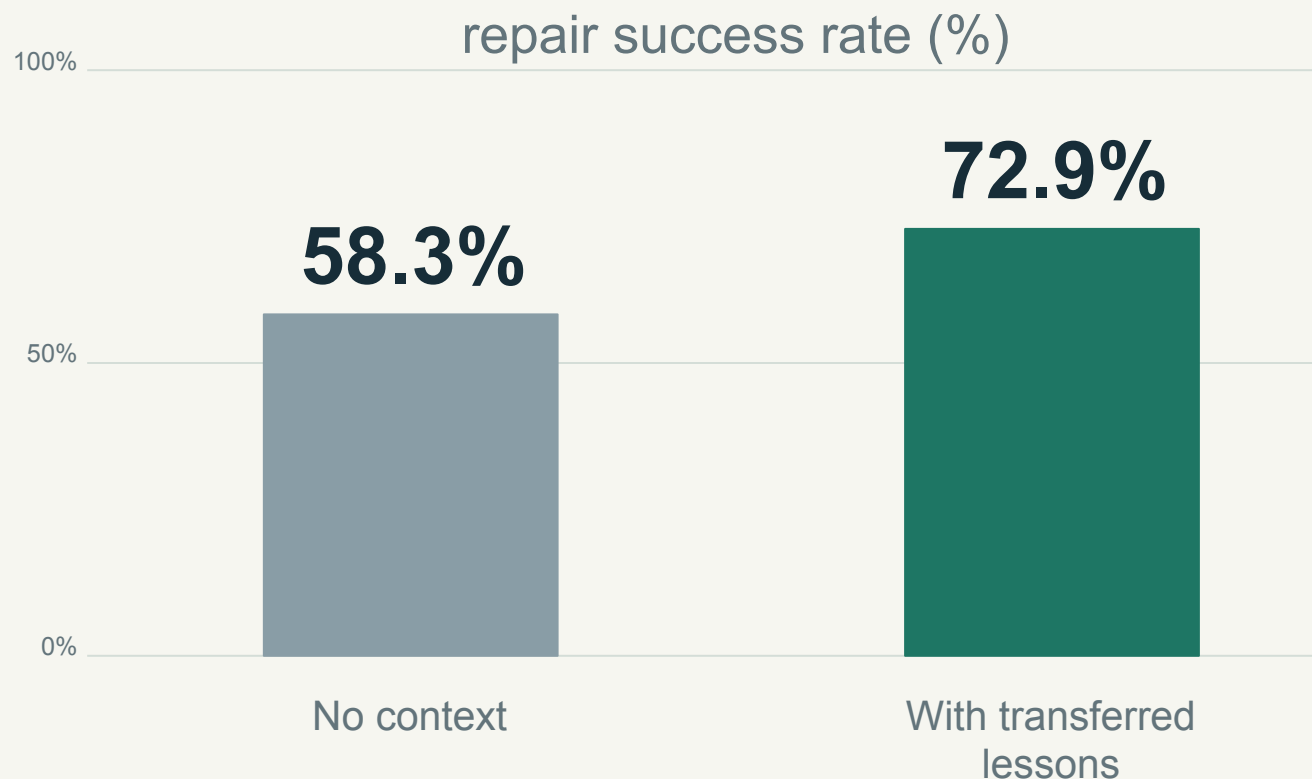
88 RANDOM TASKS

**No tested
delivery
condition beat
this baseline.**

An observed ceiling on this task sample—not a universal one.

Can decisions help on difficult tasks?

Evaluation tasks labeled “15 min–1 hour” or harder



DIRECTIONAL

+14.6 pp

25.04% increase with RCL

Less wasted exploration

Tokens per resolved task

Total tokens across evaluation attempts ÷ number of successful repairs



–32%

lower observed ratio

**Successful repairs used similar token counts;
the difference was in unsuccessful exploration.**

Observed token ratio; full authoring, maintenance, and review costs not measured.

Put the handover into the pull request.

Metatron — a reference implementation of RCL.



github.com/kerbelp/metatron

```
metatron context setup
```

AGENTS.MD · ABRIDGED CONTRACT

Read context.md
Open relevant decisions
Only then plan

ACTUAL DECISION · ABRIDGED

**In files mode, the UI
never runs git commit.**

Why? Human review remains the gate.

Working branch **→** **Human review** **→** **Shared context**

Engineering knowledge should outlive a coding session.

Repository Context Layer (RCL)

Preserve project decisions with the code.
Review them.
Let the next agent inherit them.



Explore Metatron
getmetatron.com



Website

Docs, product, and
open-source links
[Scan to explore](#)

**Thanks —
questions welcome.**

[Explore the project or connect with me.](#)

Connect with Pavel
linkedin.com/in/kerbelp



LinkedIn

Talk follow-up,
research, and updates
[Scan to connect](#)

Experimental matrix

Matched and broad task sets are different experiments.

Comparison	Executor / metric	Task set	Rows / arm*
Blind seed vs none	Local · localization	36 matched pairs × 5	180
Self vs own control	Local · localization	12 matched pairs × 5	60
Lesson transfer	Frontier · resolve / tokens	16 matched pairs × 3	48
Delivery mechanism	Local · localization / tokens	267 broad tasks × 3	801
Broad-sample ceiling	Frontier · resolve / tokens	88 random tasks × 1	88

Local: Gemma 4 (gemma4:e4b) · Frontier: Claude Opus 4.8

*Local paired rows can reuse B tasks. Self-authored lifecycle ≠ locally authored blind seed.

Complete delivery results

GEMMA 4 · 801 EPISODES PER ARM

Delivery	Localization	Context tokens / ep.	Resolve
None	48.6%	0	5.6%
Injected	51.2%	930	8.6%
File	46.8%	902	7.0%
Sharded	53.4%	305	6.7%

OPUS 4.8 · BROAD SAMPLE · 88 EPISODES PER ARM

Delivery	Resolve	Total tokens / resolved
None	90.9%	28.8K
Injected	81.8%	29.2K
File	85.2%	28.7K
Sharded	88.6%	43.8K

Validity and registration boundaries

AUTHOR COMPARISON

Capability and family confounded; self-authored lifecycle differs from blind seeding.

TOPIC CONDITIONING

Selected constraint names guide the seed; pretraining leakage is not ruled out.

MISSING OUTCOMES

Constraint-violation metric and some registered author/sequence conditions not run.

FRONTIER PRECISION

16 pairs, not ≥ 25 planned; corrected resolve $p = 0.081$; two recorded control aborts.

REGISTRATION DEVIATIONS

Analysis-plan inconsistency disclosed; delivery freeze tag private and lightweight.

UNMEASURED ARCHITECTURE

Human review, poisoning containment and concurrent merge behavior not evaluated.

Exact xarray record

pydata__xarray-4687 · local executor · repetition 1

Recorded field	No context (B)	Injected blind seed (E)
First file targeted	xarray/core/core.py	xarray/core/computation.py
First command exit code	2	0
Logged turns	14	6
Patch bytes	0	624
Prompt / completion tokens	61,387 / 3,084	23,319 / 802
Official result	Unresolved (no patch)	Unresolved

Control B later read computation.py. The example is not a causal explanation.

Exact generated diff omitted; only verified episode records and verdicts are shown.

The setup contract, from source

MANAGED AGENTS.MD BLOCK · EXCERPT

1. Run `cat context.md`
2. Open relevant decisions
3. Only then plan

Listing the directory is not reading.

DEFAULT REVIEW GATE: PR

**Author on a branch.
Review in a pull request.
Merge into shared context.**

candidate/ is optional staging.
It is never authoritative.

OPTIONAL INSERT: YOUR 20–30 SECOND OFFLINE SCREEN RECORDING

Inspect the artifact.

PAPER	<u>10.1145/3843282.3844425</u>
STUDY	<u>github.com/getmetatron/context-md</u>
IMPLEMENTATION	<u>github.com/kerbelp/metatron</u>
ARCHIVED ARTIFACT	<u>10.5281/zenodo.22122045</u>

```
cd experiment && make reproduce-paper
```

```
python paper/camera-ready-agenticdev2026/analysis/verify_claims.py
```

Commands documented in the artifact; not executed during deck preparation.