

me, my slop and i

tales of backpressure

👤 pi contributor

👤 pi-vim maintainer

github.com/lajarre/pi-vim
~1,100 installs / week

👤 how to live slop-free

👤 extra thoughts on TUIs

// contributing

■ the clanker incident

i had: clean repro · red/green tests · executable repro script

i got: PR closed

[earendil-works/pi#3930](#)

// contributing

badlogic last month

Collaborator ...

Start from a valid legacy/no-version session file containing a user message and no assistant message.

this can never happen. see SessionManager. you are also letting your clanker reply. i'm done with this issue and PR

// maintaining

■ maintainer side – price the abstraction

earendil-works/pi#4798

a well-intended request for `getEditorComponent()` composition support.

the request was itself almost certainly agent-drafted.

// maintaining

■ cost of pricing it

- **20** commits
- **3.4k** LOC delegate plumbing
- **6.7k** LOC tests
- **527** LOC e2e
- **200**-line delegation-model doc

// maintaining

■ the agent never says no.

■ the maintainer has to.

→ *say no in code. say why in writing.*

every extension point is a support contract.

// slop-aware workflow

"it is a golden era for buckets and buckets
of slop, and a dark age for gems of quality."

– **george hotz**, *the eternal sloptember* · may
2026

// slop-aware workflow



agents are upstream.



you are the backpressure.

// slop-aware workflow

■ slop defenses, inside-out

1. **authorship**
2. **repo memory**
3. **drift guards**
4. **boundary guards**

inner: human-directed. outer: automated → push outward.

```
// slop-aware workflow
```

■ 1. authorship: brain in the loop

- read the spec yourself
- review **architecture**, not just lines
- **hunk** – code and spec review
- **cmux** – agent writes QA plan, i run it

```
// slop-aware workflow
```

■ 2 and 3. repo memory + drift guards

▣ repo memory

write expectations:

- guidelines
- ADRs...

▣ drift guards

- package size
- perf benchmarks...

*pi-vim sits on the **editor critical path** → perf is critical*

```
// slop-aware workflow
```

■ 4. boundary guards: pi-garde-slop

- intercepts public-write `gh` calls
- `curl` bypasses it
- stronger → **credential boundary** (proxy)

// slop-aware workflow

■ be accountable

1. one PR, one idea
2. polish before publish
3. write your own replies
4. leave a trace of judgment

// thoughts on TUIs

■ agent TUIs are still shaped like REPLs.

■ the work is not.

// thoughts on TUIs

 demo

pi-vim · hunk · cmux

this talk was 70% agent-generated. the slop is 100% mine.

check these out:

github.com/lajarre/pi-vim | made with ❤️ @ The Nest

- **pi** → github.com/earendil-works/pi
- **pi-garde-slop** → github.com/lajarre/pi-garde-slop
- **cmux** → cmux.com
- **hunk** → github.com/modem-dev/hunk