

AGENTS.md is the new
CONTRIBUTING.md

NUTANIX

Two studies. Opposite answers.

ETH Zurich · 2026

+20% cost

AI-generated AGENTS.md files slightly lowered task success on real GitHub issues, and raised inference cost by over 20%.

Efficiency study · 124 PRs · 2026

-28.6% time

With an AGENTS.md, median agent runtime dropped by 28.6% and output tokens by 16.6%, with task completion unchanged.

Both are right. By the end of this talk, you'll know why.

ABOUT ME



MTS-3 @ NUTANIX

CKA, CKAD

Public Speaker



Agenda

- 01** The Basics
- 02** Every line is an order
- 03** Write the delta
- 04** Other agent files
- 05** Keeping it safe and fresh
- 06** Your checklist

WHAT OPEN SOURCE CONTRIBUTION TAUGHT ME

Every new contributor asks the same five questions

Build?

How do I build it?

Never?

What must I not touch?

Done?

How do I know I'm finished?

Test?

Which tests do I run?

Who?

Who reviews this?

A tiny file with a lot of adoption

60k+

open-source projects using
[AGENTS.md](#), per [agents.md](#)

- OpenAI introduced [AGENTS.md](#)
- The Linux Foundation forms AAIF with MCP, goose and [AGENTS.md](#) as founding projects
- Today: read by Codex, Copilot's coding agent, Cursor, Gemini CLI, goose, Zed, Devin and more

Polyrepo vs Monorepo

Polyrepo - one tool, one repo

kubernetes-
sigs/kustomize

kubernetes-
sigs/kind

kubernetes-
sigs/cluster-api

Each repo has its own build, tests and rules. One AGENTS.md each.

Monorepo - many projects, one repo

```
kubernetes/kubernetes
├─ cmd/kubelet
├─ pkg/scheduler
├─ staging/src/k8s.io/client-go
└─ vendor/
```

Many projects share one history. Rules change from folder to folder.

CONTRIBUTING.md vs AGENTS.md

	CONTRIBUTING.md	AGENTS.md
Reader	Humans who can ask questions	Agents that can only read and run
Tone	Welcoming, explanatory	Short, imperative, exact
Best line	"Be kind in code review"	make test WHAT=./pkg/kubelet/...
When it's missing	Contributor gets confused and asks	Agent guesses, confidently

Root for the rules, folders for the commands

```
repo/  
├ AGENTS.md ← layout, never-touch list, PR rules  
├ services/api/  
│   └ AGENTS.md ← go test ./...  
├ web/  
│   └ AGENTS.md ← pnpm test  
└ vendor/ ← never by hand
```

Nearest file wins.

It scales.

Kubernetes people know this.

Three sections earn their keep

Build commands

The exact command, copy-pasteable. Not “build the project”.

```
make WHAT=cmd/kubelet
```

Test commands

The fast, scoped one first. The full suite second.

```
make test WHAT=./pkg/kubelet/...
```

Never touch

What's generated, vendored or owned by someone else.

```
vendor/  
zz_generated*.go  
api/openapi-spec/
```

If a line isn't a command or a boundary, ask why it's there.

Every line is an order

+14–22%

more reasoning tokens
with a context file

+2–4

extra steps per task

160×

more use of a tool just
because the file named it

Inflation

- Project overviews
- Directory tours
- Style essays
- Copies of the docs

→ **more work**

Compression

- Exact build commands
- Scoped test commands
- Never-touch paths
- Slow tests to skip

→ **less work**

An illustrative AGENTS.md for a Kubernetes-style repo

```
# AGENTS.md
## Build
- One binary: `make WHAT=cmd/kubelet`
## Test
- Scoped: `make test WHAT=./pkg/kubelet/...`
- Before a PR: `hack/verify-gofmt.sh`
## Never edit by hand
- vendor/ → run `hack/update-vendor.sh`
- zz_generated/*.go → run `hack/update-codegen.sh`
- api/openapi-spec/ → run `hack/update-openapi-spec.sh`
## PRs
- One logical change per PR
- Fill in the release-note block
```

About 15 lines.

Every line is a command to run or a place not to go. Nothing the agent could learn from the code itself.

Same intent, rewritten for a reader who can't ask

Before - a typical first draft

```
# Guide for AI agents  
This project is a fast, cloud-native  
platform for... (copied from README)
```

```
Please write clean, idiomatic Go  
and be careful with changes.
```

```
Follow our style guide.
```

```
To test, run `make test-all`
```

After - the same file, 8 lines

```
## Overview  
- See README.md  
## Test  
- `make test WHAT=./pkg/<pkg>/...`  
## Style  
- `hack/verify-gofmt.sh` must pass  
## Never edit by hand  
- vendor/ → `hack/update-vendor.sh`  
- zz_generated.*.go → `hack/update-codegen.sh`
```

Copy → link. Advice → command. “Be careful” → a boundary. Stale command → one you ran this week.

Write the delta, not the docs

What the agent needs

- what it already knows
- what's already in the repo

= **your AGENTS.md**

When researchers removed a repo's docs first, AI-generated files improved by 2.7% and beat human-written ones.

The line ablation test

- 1 Delete one line from AGENTS.md
- 2 Rerun the same small task
- 3 No change? The line was only costing tokens

One source of truth, thin pointers everywhere else

CLAUDE.md

.github/copilot-instructions.md

.cursor/rules/



AGENTS.md

Build, test, boundaries. Written once.

Three layers of agent context

AGENTS.md

The repo. How to build, test, and what never to touch. Always loaded.

Skills

The job. A playbook for one task, loaded only when that task comes up.

MCP

The tools. Live access to systems like issue trackers, logs and docs.

Three numbers, all from the PR history

Metric	What to count on agent PRs	Beyond code: tickets
Acceptance rate	Agent PRs merged $\div$ opened	Suggestions used $\div$ suggestions made
Right first time	First CI run is green	First answer correct, no re-prompting
Human edit count	Human commits on the agent PR	Edits to the draft before it's used

AGENTS.md is an attack surface

Instructions that run

Agents with shell access carry out what the file says. It's closer to CI config than to docs.

One PR changes every agent

Edit AGENTS.md and you change what every agent does in the repo, for every task.

Nested files count

Nearest file wins. Check whether vendored or third-party folders can carry their own.

Stale instructions are the new stale dependencies

```
# idea: a CI check for AGENTS.md
on: pull_request
steps:
  1. extract every `command` from AGENTS.md
  2. run each one in a clean container
  3. fail the PR if any command breaks

# and treat the files like artifacts
- an owner in CODEOWNERS
- reviewed in PRs
- a line in the changelog
```

Do this in your repo next week

1. Copy the build and test commands from CI into [AGENTS.md](#)
2. List everything that must never be edited by hand
3. Point CLAUDE.md and copilot-instructions.md at it
4. Give it a “Never edit by hand” section
5. Track acceptance, right first time and human edits

Questions?

Agents are just the newest contributors. Onboard them like you mean it.

Thank You!



Unnati Mishra

unnaticse2019@gmail.com

linkedin.com/in/pingunnatimishra

X- @ping_Unnati



AGENTS.md is the New CONTRIBUTING.md: A Field Report from One
Year of Agent-Friendly Monorepos

Feedback