



Skills & MCPs for Drupal Development

Alex Urevick-Ackelsberg – Zivtech | DrupalCamp NJ 2026

WHO AM I?

- Business owner — **Zivtech** (Drupal agency, Philadelphia)
 - Also: **Milk Jawn** (ice cream manufacturer + retailer, Philadelphia)
- Strategist and site builder
- Long-time Drupal-ist (First Drupal site: Young Philly Politics — Drupal 4.6)
- Used to contribute code, but stopped when I partnered with / hired people who were actually good at that
- **Claude Code changes that**

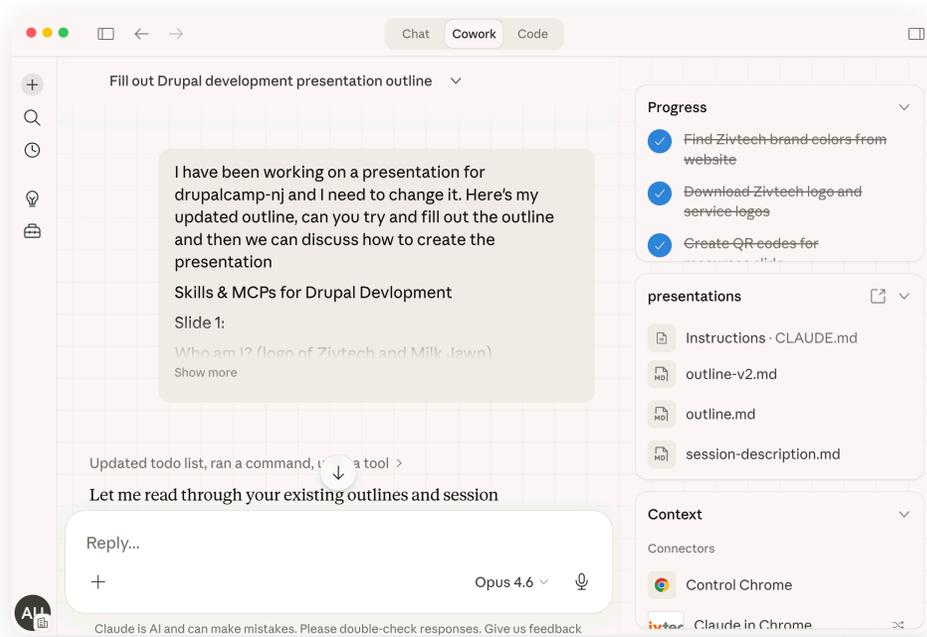


CLAUDE CODE

- Work in a **git repository** – reads files, understands project structure, navigates codebases
- Writes its own helper applications when needed
- Can easily leverage **Skills** and **MCP servers**
- Really adept at using **multiple agents** – spawns sub-agents for parallel tasks



CLAUDE COWORK



- Gives much of the **power of Claude Code** in a desktop app
- File management, document creation, browser automation
- Same Skills and MCP concepts — same ecosystem
- To really get the full experience, install **git**
- Worth getting set up in **Claude Code** if you can
- Cowork is the accessible on-ramp; Claude Code is where the serious development happens

AI coding assistants are
powerful but **generic**

"They don't know Drupal's conventions"

You end up babysitting every response

WHAT ARE SKILLS?

"Skills are markdown files that teach AI how you work."

- A folder containing a `SKILL.md` file
- Written in plain Markdown – no SDK, no special syntax
- Claude reads it before performing related tasks
- Think of it as: giving a brilliant but new hire the team handbook on day one

*Skills encode that knowledge as both **enablers** (teaching AI what to do) and **guardrails** (stopping it from doing what you don't want).*

SKILLS — WHERE THEY LIVE

```
your-project/
├── .claude/
│   └── skills/
│       ├── drupal-coding-standards/
│       │   └── SKILL.md
│       ├── local-dev-setup/
│       │   └── SKILL.md
│       └── ticket-writing/
│           └── SKILL.md
```

Lives in `.claude/skills/` — automatically available when working in that repo

WHY THIS LOCATION?

Claude Code scans `.claude/` on startup. Skills in this folder are discovered automatically — no configuration required.

SHAREABLE

Commit the `.claude/` folder. Every developer on the team gets the same AI behavior.

SKILLS — WHAT GOES IN THEM

CODING STANDARDS

"Use `elseif` not `else if`. Two-space indentation."

DEVELOPMENT WORKFLOWS

"Branch naming is `feature/PROJ-123-short-description`."

ARCHITECTURE DECISIONS

"Use Paragraphs for flexible content."

BUSINESS RULES

"Free shipping over \$50."

DOMAIN EXPERTISE

"Cache tags must be invalidated when entities are updated."

CORRECT AND INCORRECT EXAMPLES

Claude learns from contrast. Show what *not* to do alongside what to do.

Skills encode organizational knowledge — making tribal knowledge machine-readable and shareable

WHAT IS MCP?

Model Context Protocol

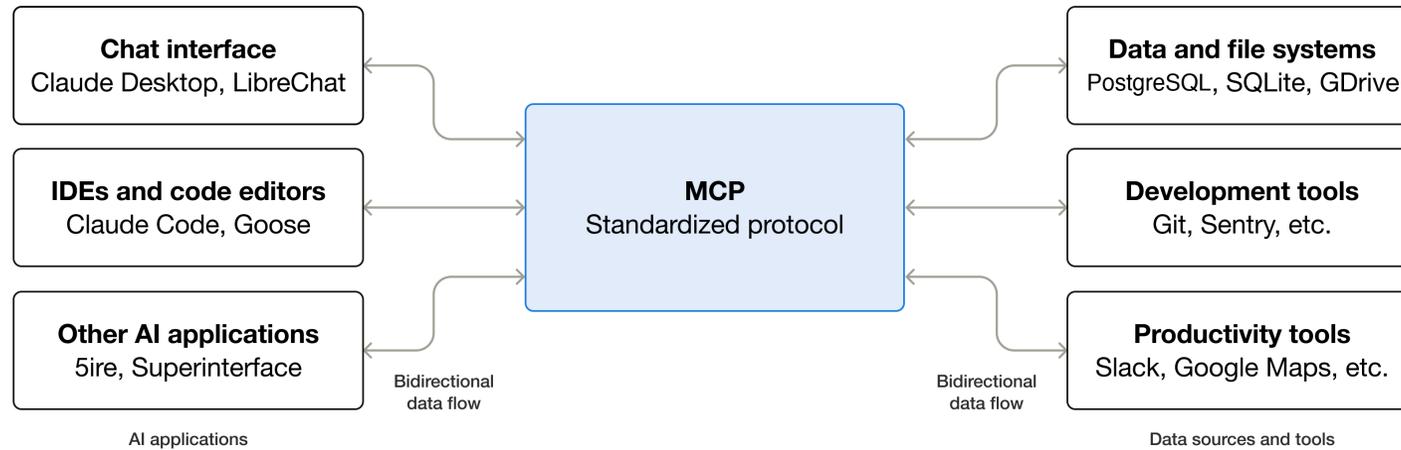
An open standard from Anthropic



- A standardized way for AI to connect to real tools and services
- **"USB for AI"** – any AI client can talk to any MCP server
- Build one server, works everywhere

Skills give Claude the knowledge. MCP gives Claude the capabilities.

MCP — HOW IT WORKS



Source: modelcontextprotocol.io

Tools
actions the AI can take

Resources
data the AI can read

Prompts
reusable interaction patterns

MCP — SUPPORTED CLIENTS

Claude Code
+ Claude Cowork

Cursor

VS Code
(GitHub Copilot)

ChatGPT / Codex

Windsurf

JetBrains IDEs

And growing...

Your MCP servers work across all of them. **Build once, use everywhere.**

MCPS I USE — DEVELOPMENT

| MCP Server | What It Does | How I Use It |
|-------------------|--|---|
| Playwright | Browser automation — navigate, click, fill forms, screenshots, a11y audits | Automated a11y testing with axe-core. Test UIs. Scrape content for migrations. |
| Atlassian | Jira + Confluence — read/write tickets, update statuses, search docs | Claude reads a Jira ticket, does the work, updates the ticket status |
| GitHub | Repos, issues, PRs — create branches, open PRs, review code | Full PR workflow — Claude creates branch, commits, opens PR, responds to review |

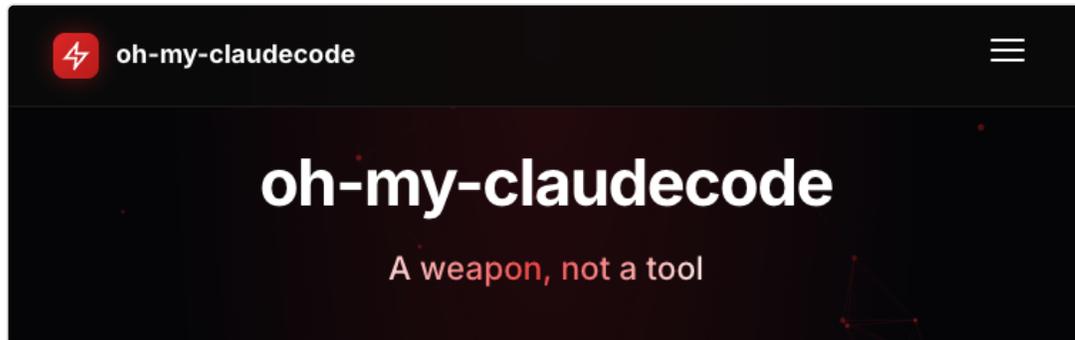
MCPS I USE — BUSINESS & OPERATIONS

| MCP Server | What It Does | How I Use It |
|---------------|---------------------------------------|---|
| Google | Drive, Gmail, Analytics | Document management, email drafts, analytics |
| Pencil | AI-assisted UI design and prototyping | Component designs, layouts, mockups |
| Square | Point-of-sale + commerce | Milk Jawn retail — sales reports, inventory |
| Otter | Meeting transcripts | "What did the client say about the timeline?" |
| AWS | Cloud infrastructure | Server management, log analysis, deployment |

OTHER USEFUL HELPERS

OH-MY-CLAUDECODE (OMC)

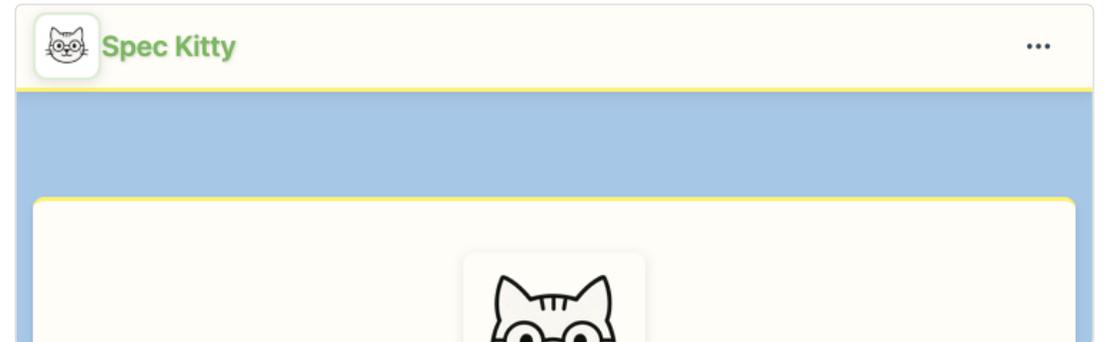
- Open-source multi-agent orchestration layer
- Team mode pipeline & parallel agent execution
- Deep interviews: Socratic questioning
- Cross-review: different model reviews



github.com/Yeachan-Heo/oh-my-claudecode

SPEC-KITTY

- Reverse-engineers existing websites into specs
- Plans new features by analyzing what's built
- Great for onboarding or planning migrations



github.com/Privity-ai/spec-kitty

THE IMPROVEMENT WORKFLOW

PLAN PHASE

1 Plan — define what you're building



2 Criticize — harsh review of the plan



3 Revise — improve until solid

Loop steps 2-3 until the plan is right

EXECUTE PHASE

4 Execute — implement the plan



5 Criticize — review the execution



6 Revise — fix and improve

Loop steps 5-6, then back to step 1

This pattern is baked into every skill we build. Next: the skills that encode it.

HARSH-CRITIC

Based on **oh-my-claudecode's critic mode** — a 5-phase review protocol that ensures AI doesn't just validate your assumptions.

1. **Pre-commitment predictions** — state what you expect before looking
2. **Evidence gathering** — verify with actual code/data
3. **Multi-perspective review** — examine from multiple lenses
4. **Gap analysis** — what's missing?
5. **Synthesis** — severity-rated findings

Perspectives: Audience, Stakeholder, Accessibility, Security, Performance, Maintainability — each surfaces different blind spots.

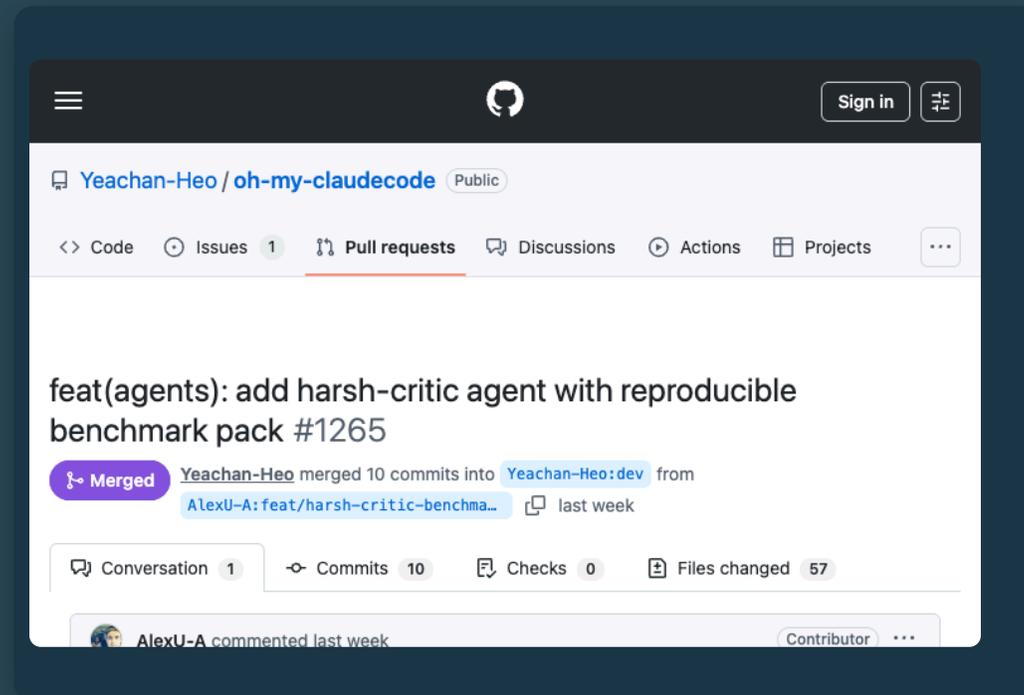
33

gap items found
vs. 0 with standard AI review

SIDENOTE

ALWAYS BE IMPROVING

harsh-critic started as a standalone skill. I contributed it to oh-my-claudecode where it became **the** critic mode — used by thousands of developers.



First it was its own critic. Now it *is* the critic.

THE PROBLEM — DRUPAL EXAMPLES

CODING & ARCHITECTURE

- `else if` instead of `elseif`
- Suggesting custom code when a contrib module exists
- Missing cache tag invalidation
- Creating entity types when content type + paragraphs would do
- Ignoring hook naming conventions & annotation syntax
- Not knowing which test type to use (Unit vs Kernel vs Functional)

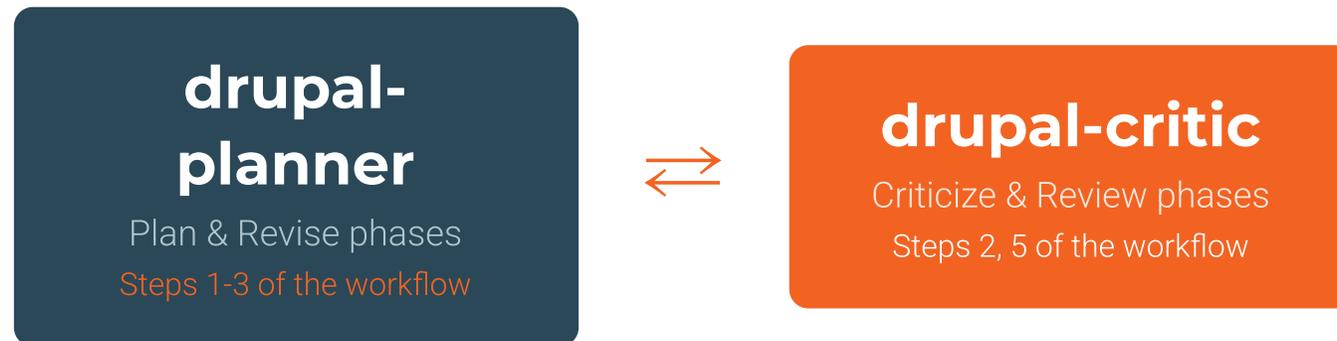
COMMUNITY & CONTRIBUTION

- Not searching Drupal.org issue queues before writing fixes
- Duplicating work by ignoring existing MRs & patches
- Local fixes that vanish on `composer update`
- Creating low-quality issues that burden maintainers
- Not knowing MR-based vs patch-based contribution workflows
- Writing tests that validate code logic but miss user experience

ENCODING THE WORKFLOW INTO SKILLS

The improvement workflow isn't just a process — it's **architecture**.

We built Drupal-specific skills for each phase:



Together they encode the full **Plan** → **Criticize** → **Revise** → **Execute** → **Criticize** → **Revise** cycle for Drupal projects.

DRUPAL-PLANNER

A meta-skill that plans Drupal projects by calling specialized sub-skills

drupal-planner (core)

Orchestrates sub-skills based on project scope, delegates to specialized planners

content-model-planner

Entity types, bundles, paragraphs, field architecture, editorial workflows

taxonomy-planner

Vocabularies, term hierarchies, faceted navigation, Views integration

search-discovery-planner

Search API, Solr / Elasticsearch backends, facets, autocomplete

a11y-planner

Accessibility standards, ARIA patterns, keyboard navigation, screen readers

drupal-migration-planner

Migration sources, process plugins, rollback strategies, incremental sync

DRUPAL-CRITIC

The review counterpart – orchestrates 24 community-built external skills

drupal-theme-critic

Base theme selection, SDC patterns, CSS methodology, Twig templates

content-model-critic

Entity proliferation, field reuse, editorial workflow gaps

taxonomy-critic

Hierarchy depth, naming conventions, cross-vocabulary overlaps

search-discovery-critic

Index bloat, relevance tuning, facet performance

a11y-critic

WCAG compliance, contrast ratios, focus management

perf-critic

Cache strategy, render pipeline, database query patterns

Started as harsh-critic adapted for Drupal, now expanded to orchestrate 24 domain-specific sub-skills covering architecture, security, accessibility, performance, and more.

WORKING WITH DRUPAL

Building Drupal sites — planning and reviewing with specialized sub-skills.

PLAN

content-model-planner

Entity types, bundles, paragraphs

taxonomy-planner

Vocabularies, hierarchy, facets

search-discovery-planner

Index design, facets, relevance

CRITICIZE

content-model-critic

Entity proliferation, field reuse

taxonomy-critic

Hierarchy balance, editorial usability

search-discovery-critic

Zero-result handling, relevance tuning

*Each planner has a matching critic. Next: the meta-skills that orchestrate them — **drupal-planner** & **drupal-critic**.*

WORKING ON DRUPAL

Contributing back to the Drupal community – not just building with it.

CONTRIBUTE

drupal-contribute-fix

Search upstream first, then fix & package

drupal-intent-testing

Before/after browser-based verification

drupal-coding-standards

PHPCS compliance & conventions

VALIDATE

drupal-intent-testing

Run baseline → apply fix → compare

drupal-coding-standards

Hook format, annotation syntax, tests

drupal-critic

Full review before submitting MR

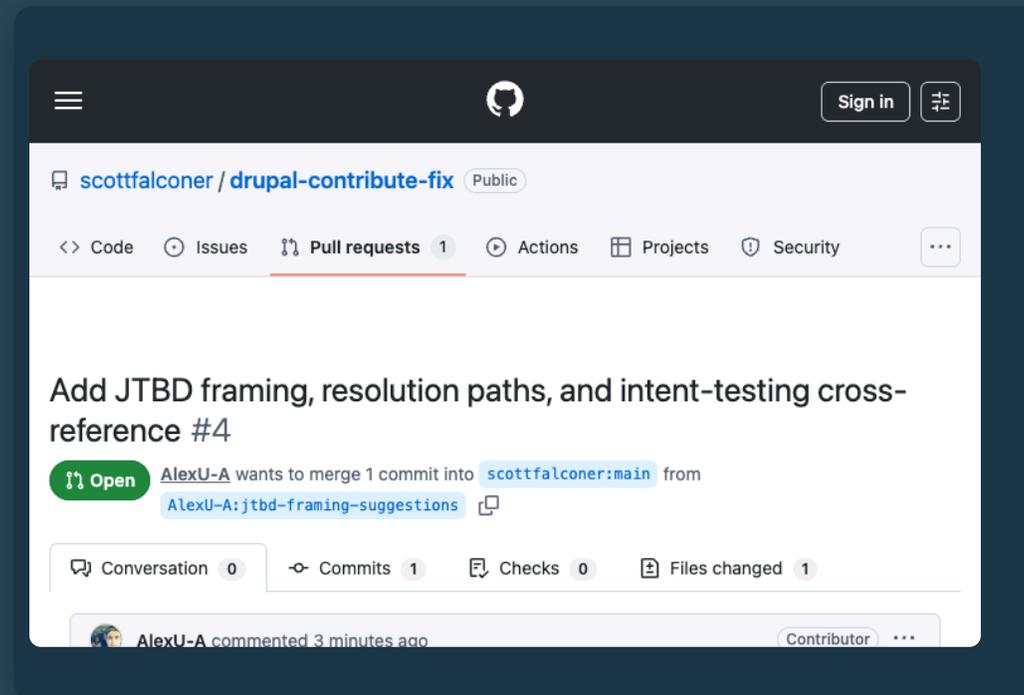
The same Plan → Criticize → Revise workflow, applied to open-source contribution.

SIDENOTE

ALWAYS BE IMPROVING

While analyzing these skills for the talk, Claude noticed they could benefit from JTBD framing.

So we filed PRs. During the presentation prep.



Improve the tools you use, as you use them.

OTHER USEFUL SKILLS I LOVE

FIND-SKILLS

Discovers & recommends skills from the community

SKILL-CREATOR

Guides you through building new skills

FRONT-END-DESIGN

UI design, responsive layouts, components

VISUAL-EXPLAINER

Generates visual diagrams from code or concepts

▲ / Skills

Audits

Docs

SKILLS

Community skill registry – skills.sh

RESOURCES

- **Claude Code** docs.anthropic.com/.../claude-code
- **Claude Cowork** claude.ai/cowork
- **Skills Docs** docs.anthropic.com/.../skills
- **MCP Specification** modelcontextprotocol.io
- **skills.sh** skills.sh
- **oh-my-claudecode** github.com/Yeachan-Heo/oh-my-claudecode

- **Zivtech Claude Skills** github.com/zivtech/zivtech-claude-skills
- **harsh-critic / drupal-critic** github.com/zivtech/harsh-critic
- **spec-kitty** github.com/Priivacy-ai/spec-kitty
- **drupal-contribute-fix** github.com/scottfalconer/drupal-contribute-fix
- **drupal-intent-testing** github.com/scottfalconer/drupal-intent-testing
- **Playwright MCP** github.com/anthropics/mcp-playwright

zivtech

Q&A

Alex Urevick-Ackelsberg

alex@zivtech.com | zivtech.com