



STIR TREK 2026 | FRIDAY, MAY 1

Claude Code in Action

Transforming Custom Software Development with Agentic AI

Kevin Grossnicklaus · ArchitectNow

A B O U T

Hi, I'm Kevin.

Founder & President, ArchitectNow.



A B O U T

- ▶ First line of code in 1986 — Apple IIgs, Woz LE.
- ▶ 18 yrs running ArchitectNow.
- ▶ Microsoft Partner · AI Apps on Azure.
- ▶ Remote team — US + Brazil.
- ▶ Most days I'm building the stuff in this talk.

F I N D M E

**LinkedIn**

scan →

GitHub

@kvgros2

Email

kvgros@architectnow.net

architectnow.net

WHAT IT IS

Claude Code in 30 seconds.

WHAT IT IS

Anthropic's tool for agentic software development. Claude doesn't just suggest code — it plans, edits files, runs your shell, and verifies its own work.

WHERE IT RUNS

Primary surface is the command line in any repo. Also inside Claude Desktop and the Claude mobile app.

WHAT YOU NEED

A paid Anthropic account. Pro ~\$20/mo · Max 20x ~\$200/mo.
API / Bedrock / Vertex for programmatic and CI use.

WHAT POWERS IT

Anthropic's frontier models — Opus for heavy reasoning, Sonnet for fast/default work, Haiku for lightweight tasks. Pick per session with /model.

Setup. Usage. Extension. Resources.

- 01 Install Claude Code and run your first task
- 02 Real-world usage patterns
- 03 SpecKit + MCPs + extending the agent
- 04 Where to go next — the slide you'll want to download

VENUE HEADS-UP

No Wi-Fi, no cell. Slide-heavy and link-rich on purpose — less live, more take-home.

MENTAL MODEL

Plan → ***Act*** → ***Observe*** → ***Repeat.***

The spec is the durable artifact.

- ▶ You stay in the chair. Claude plans, edits, runs, verifies. You review and direct.
- ▶ When implementation gets cheap, the spec wins. That's the second half of this talk.

S E T U P

Three commands. You're in.

```
npm install -g @anthropic-ai/claude-code  
claude -chrome  
/login
```

Auth in the browser on first run.

Pro, Max, and Console accounts all work.

Run claude in any repo. That's the entire setup.

S E T U P

Three things to do before you ask it for anything.

`/init`

Claude Code reads the repo and proposes a CLAUDE.md.

Edit it. Commit it.

`/permissions`

Allow-list shell commands you trust.

Lock down the rest.

`/model`

Switch between Sonnet (fast) and Opus (heavier reasoning).

Three commands. Two minutes. The repo is now configured for the rest of the talk.

SETUP

***CLAUDE.md** — the highest-leverage file in your repo.*

Tell it once. Reuse forever.

▶ **Project type**

Primary language, framework, runtime.

▶ **How to run**

Build, test, lint, dev-server commands.

▶ **Conventions**

Naming rules, folder layout, code style.

▶ **Guardrails**

Always do this. Never do this.

▶ **Spec location**

Where the team's specs live (Notion, /docs, ADRs).

▶ **Stack notes**

External services, auth, env vars to know.

Live walkthrough — local CLAUDE.md from a real ArchitectNow repo. No network.

Claude doesn't just write code. It runs your shell.

Stop thinking "AI assistant." Start thinking junior engineer with shell access who never gets tired.

WHAT "TOOL" MEANS HERE

- ▶ File edits, deletes, moves
- ▶ Git commits, branches, tags
- ▶ npm · pip · dotnet · cargo
- ▶ gh · az · gcloud · kubectl
- ▶ ripgrep · jq · curl
- ▶ Your team's custom scripts

PERMISSIONS ARE NOT OPTIONAL

- ▶ /permissions allow- and deny-list per repo, per session.
- ▶ Azure CLI, gcloud, kubectl, gh inherit YOUR login. Be deliberate.
- ▶ PreToolUse hooks (Section 6) add a programmatic guardrail.

The trick to agentic AI isn't the model — it's giving it tools and trusting it to pick the right one.

"Add this feature. Here's the spec."

- ▶ Drop the spec in chat or reference a file.
- ▶ Claude reads relevant code, proposes a plan, asks clarifying questions.
- ▶ You approve or correct. It implements, runs tests, and reports back.
- ▶ Time: a typical 30-min feature in 5–10 min of human attention.

REAL EXAMPLE

Webhook signature verification middleware on a .NET API — added in 8 minutes including tests.

Where Claude Code is most decisive.

Refactor + test backfill on legacy code.

- ▶ Point it at the file or module. Ask for the refactor with explicit constraints.
- ▶ It can generate tests FIRST against current behavior — then refactors — then re-runs.
- ▶ This is the killer use case. Code nobody wanted to touch is now low-risk.

REAL EXAMPLE

1,400-line legacy controller refactored into testable services with 85% test coverage in an afternoon.

Feed it the bug. Watch it work the loop.

- ▶ Paste the stack trace, link the failing test, or just describe the symptom.
- ▶ Claude reproduces, reads logs, narrows the root cause, proposes the fix.
- ▶ You approve. It fixes, adds the regression test, opens the PR.
- ▶ This is also where teams realize Claude Code can run unattended on background work.

S P E C K I T

GitHub SpecKit. Free. Drop it in any repo.

```
uvx --from git+https://github.com/github/spec-kit.git \  
specify init <project>
```

- ▶ Open-source SDD scaffolding from GitHub. Model-agnostic.
- ▶ Drops a .specify/ folder with templates, scripts, and command shortcuts.
- ▶ Works with Claude Code, Copilot, Cursor, Gemini — same artifacts, any agent.
- ▶ Once installed, the workflow runs through /speckit.* slash commands inside Claude Code.

github.com/github/spec-kit

S P E C K I T

Specify → ***Clarify*** → ***Plan*** → ***Tasks*** → ***Implement***.

/speckit.specify

Write what and why.

Output: spec.md

/speckit.clarify

AI evaluates and asks structured questions on edge cases and ambiguity.

Output: tightened spec.md

/speckit.plan

Pick the tech approach against the clarified spec.

Output: plan.md

/speckit.tasks

Break the plan into discrete, parallelizable work.

Output: tasks.md

/speckit.implement

Claude Code executes against the locked artifacts.

Output: code + tests

/speckit.clarify is where SpecKit earns its keep — AI as the spec reviewer surfaces edge cases humans miss.

S P E C K I T

Less typing. Better software. Reviews shift.

- ▶ Iterate the spec WITH Claude. The spec gets tight before code is written.
- ▶ Lock the spec. Hand it off. Implementation, tests, and as-built docs come back together.
- ▶ Reviews change: instead of reading 800 lines of code, you verify the artifacts trace back to the spec.

REAL EXAMPLE

Greenfield internal tool: spec'd in a morning, implemented and tested by Claude that afternoon, in production by Friday.

SPECKIT · SPEC VS VIBE

Not every change needs a five-stage spec.

ONE-SHOT / VIBE

< 30 min · low stakes

You review the output yourself.

Throwaway scripts, doc updates, prototyping.

STRUCTURED PROMPT

1–2 hrs · others review

Multi-paragraph prompt with goals + constraints.

Captures intent without producing artifacts.

FULL SPECKIT SDD

Multi-day · auditable

Documentation per stage, human review at each gate.

Larger features, modernizations, anything someone else extends.

SAME PROMPT, TWO PATHS

Evaluate all code in this repo. Update /docs as markdown, update README.md to current state, refresh CLAUDE.md + Speckit constitution. Use Mermaid for diagrams.

VIBE: paste, hit go, review.

Fast. Repos you know cold.

SDD: wrap in /speckit.specify.

Auditable. Reviewable. Slower.

EXTENDING · MCPS

If Claude Code can't reach it, it can't help with it.

MCP = open standard for connecting AI to tools. Build the server once; every model uses it.

```
claude mcp add github --transport stdio -- npx @github/mcp-server
claude mcp list
```

WHAT WE WIRE UP AT ARCHITECTNOW

- ▶ GitHub / Azure DevOps — read tickets, search code, open PRs
- ▶ Azure MCP — provision and manage Azure resources, ARM/Bicep flows
- ▶ Figma — pull designs, frames, and tokens into context for UI work
- ▶ Notion — pull specs and as-built docs into context
- ▶ Postgres / SQL Server — query staging data live during dev
- ▶ Custom MCPs — internal CRM, time tracking, our own services

E X T E N D I N G

Build it yourself. Or grab it from the ecosystem.

Slash commands

`.claude/commands/*.md`

Encoded workflows. /pr-review, /spec-new. Team playbook on day one.

Subagents

`.claude/agents/*.md`

Specialist agents — reviewer, test-writer, security. Run in parallel.

Hooks

`.claude/settings.json`

Guardrails on tool use, prompt submission, session stop. Block, lint, log.

Plugins

agents + commands + hooks + MCPs

Bundle and distribute org-wide. Install with one line.

Build your own with Claude Code itself. · Or grab from the ecosystem — Awesome Claude Code, ClaudeLog, MCP directory.

PRODUCTION

Embedded, not bolted on.

- ▶ Coding, refactoring, modernization
- ▶ Spec writing and stress-testing
- ▶ As-built doc generation and maintenance
- ▶ Test generation and backfill
- ▶ DevOps, GitHub Actions, Azure provisioning
- ▶ Pull request reviews (CodeRabbit + Claude)
- ▶ Standardization via plugins and hooks
- ▶ Data access and analysis against staging databases

The deck is a takeaway — every command, config, and link on this slide is reusable Monday.

PRODUCTION · CLAUDE CODE ANYWHERE

Your terminal isn't the only place Claude lives.

NATIVE

macOS, Windows (WSL2), Linux. Same CLI, same models, same plugins everywhere.

MOBILE & TABLET

Claude apps on iOS and Android. Claude Desktop on macOS and Windows. Review, light edits, kick off long runs.

REMOTE CONTROL

`/remote-control` drives a Claude Code session from another device or browser. Start a task on your laptop, monitor it from your phone.

CLOUD SESSIONS

Run Claude Code in a Linux VM in Azure / AWS / GCP. Isolated environment, separate credentials, tighter blast radius.

ArchitectNow pattern: several developers run dedicated cloud VMs as their primary Claude Code environment.

PRO TIP · WISPR FLOW

Stop typing your prompts.

The fastest way to write a complex prompt is to say it out loud.

WHAT IT IS

Wispr Flow — AI-powered voice-to-text. Hold a hotkey, speak, release. Cleaned, punctuated text drops wherever your cursor is.

WHY FOR CLAUDE CODE

Long prompts, specs, refactor briefs, reviews — faster spoken than typed. Stream of consciousness becomes a structured prompt.

wisprflow.ai · macOS · Windows · iOS · iPadOS

The Claude Code use case sells it. The rest of your day pays for it.

RESOURCES

Get the deck. Get the links.



architectnow.net/downloads

Scan it. Bookmark it. Share it Monday.

RESOURCES

Where to go from here.

START HERE

Claude Code docs

docs.claude.com/claude-code

Best Practices

anthropic.com/engineering

SpecKit

github.com/github/spec-kit

EXTEND

Hooks docs

docs.claude.com/claude-code/hooks

Subagents docs

docs.claude.com/claude-code/sub-agents

Plugins docs

docs.claude.com/claude-code/plugins

PLUG INTO YOUR STACK

MCP spec + servers

modelcontextprotocol.io

GitHub MCP

github.com/github/github-mcp-server

Azure DevOps MCP

github.com/microsoft/azure-devops-mcp

Azure MCP · Figma MCP

github.com/Azure/azure-mcp

PRODUCTIVITY + COMMUNITY

Wispr Flow (voice-to-text)

wisprflow.ai

Awesome Claude Code

github.com/hesreallyhim/awesome-claude-code

ClaudeLog

claudejlog.com

This deck + extras

architectnow.net/stirtrek



CLOSING · WE LIVE IN MAGIC TIMES

Agentic dev isn't going anywhere. Lean in.

The technical floor rose. The judgment ceiling didn't.

STAY IN TOUCH



Kevin Grossnicklaus

Founder & President, ArchitectNow

kvgros@architectnow.net

GitHub: @kvgros2

LinkedIn: scan ←

TAKE IT HOME



Deck + every link

architectnow.net/downloads

B O N U S



Nobody is immune.

Dingle Peninsula, April 2026.