Coding with Claude

6 months with an AI Agent

Karl Matthias
Member of Technical Staff - Mozi

Who Am I

Nearly 30 years in tech, 10 startups

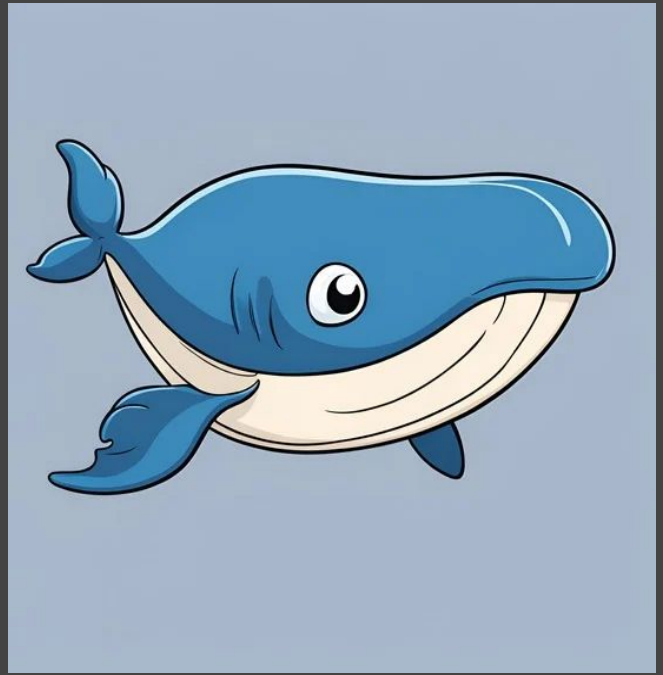
ex-New Relic, ex-Nitro, ex-InVision, ex-Community.com (etc)

Co-Wrote *Docker: Up and Running* from O'Reilly Media (3 editions)

Working remote since 2018

In Ireland since April 2016

Coding every day



Tom Patterer

Tom did a lot of this learning!

- Based in Vienna
- Member of Technical Staff at Mozi
- ex-Community, co-founded a couple of startups (RecordBird, Sendmate)
- Awesome dude

This Talk

What Mozi engineers have learned from 6 months with Claude

Why we do what we do

Reflects previous 6 months — things are moving fast

Practical focus, no real theory here

Everyone has to solve the same problems

Happy to answer questions at the end

Journey of 6 Months

How it started

1. Short requests, not super specific
2. Write whole spec
3. Write request to Claude to make the plan

How it’s going

4. Have Claude make the plan, generate sequence diagrams, docs

Your job:

Get Claude to do the work, you review

Effective Agentic Coding

Setting the Agent up for success is your responsibility. You're a team. AI agents are non-deterministic.

CLAUDE.md is almost magic (Use ~/.claude/CLAUDE.md too)

- Specify which libraries to use, which patterns to use
- Any time it does something wrong, put it in there
- Get it to write the rules for itself by explaining the mistake

Code Analysis

The better Claude understands your code, the better the output

- Have it analyze your code for patterns
- Put good ones in CLAUDE.md
- Have it refactor bad ones

Tell it about your system architecture

- Worst case, let it scan all your repos

Help Claude Out

Use subagents!

Clear the context!

Install the tools it may need to access your data, PRs, pull requests, etc

Pair it with Gemini CLI for inspection

Good Practices Stack Up

What's good for humans is good for Claude (garbage in, garbage out, too)

- Format your code!
- Have and maintain good tests!
- Have a Makefile or script directory with all the common commands
- Write code generation tools
- Standardize your code as much as possible
- Use schemas!
- Work in chunks, not all at once

Caveats

The slower your test cycle, the worse your AI experience will be

You MUST have good tests

Some languages work better than others (E.g. Go is great, Elixir is very good, Swift is just OK)

Permissions

Claude will constantly nag you for permissions – tedious at first

--dangerously-skip-permissions is possible, but well, dangerous

Talk to Roland about sandboxing!

Great First Use Cases

Refactor old pattern into a new one

Look at an existing commit and repeat it on a different codebase

Identify antipatterns in your code with explanations

Translate code from one language to another

Building Libraries

Use commands, they are simple

Have Claude write tools

MCP is more complex, but can help

Skills are new, may replace a lot of MCP servers

Demo

Where to Find Me

Blog: relistan.com

Mastodon: @relistan@k.matthias.org

LinkedIn:
https://www.linkedin.com/in/karlmatthias/

Book: *Docker: Up and Running (3rd ed)*

Mozi: https://mozi.app