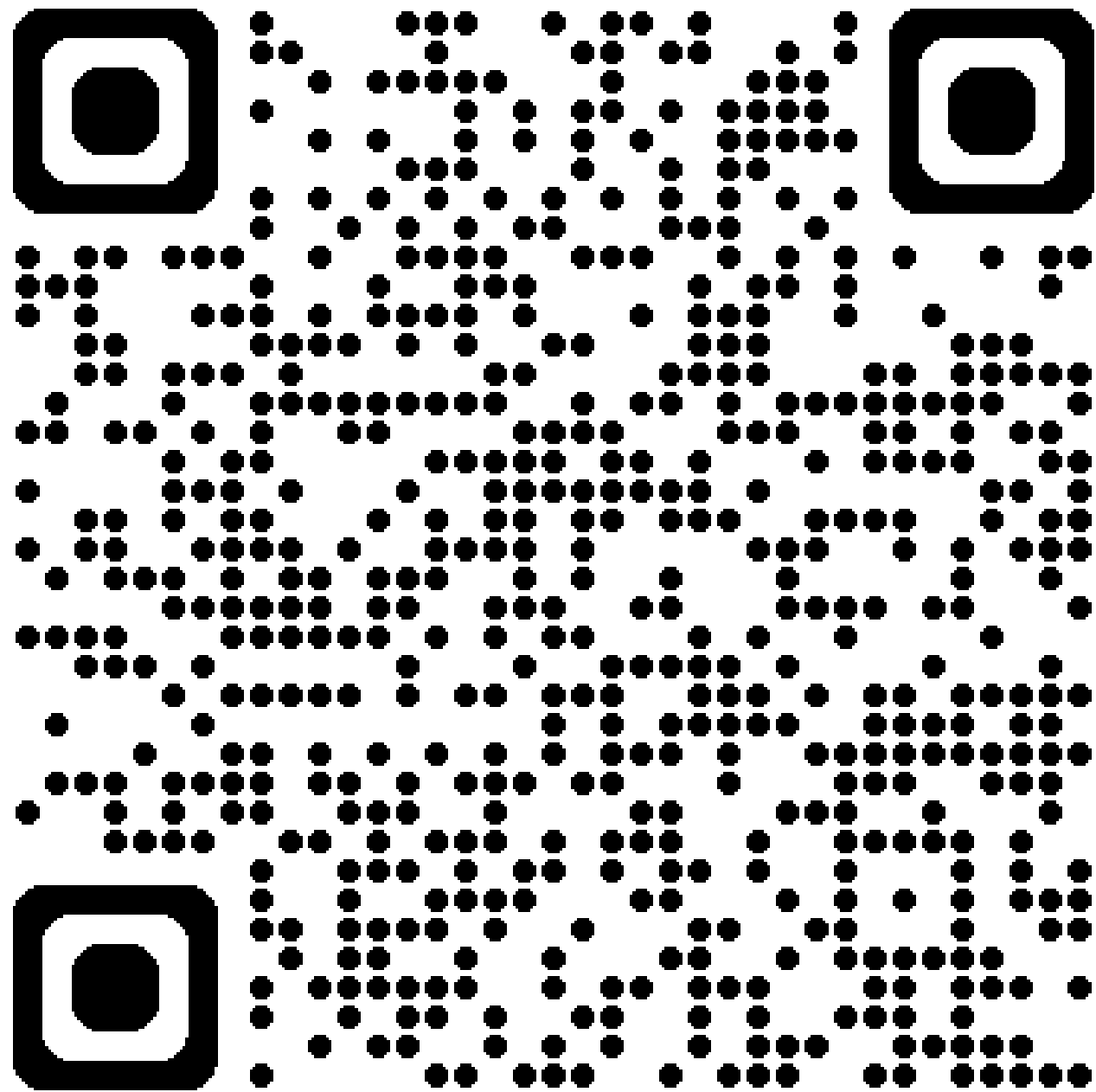


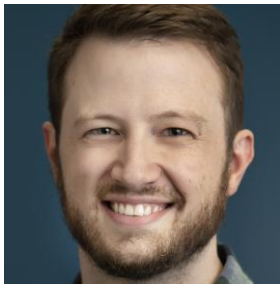
Zero to Hero with GitHub Copilot

What Works in Real Projects

Daniel Ward



Me



LEAN
TECHNIQUES



- Software developer, consultant
- Microsoft .NET MVP
- Co-organizer of the San Antonio/Austin .NET User Group
-  daninacan.com
-  [@danielwarddev](https://twitter.com/danielwarddev)
-  [daniel-ward-dev](https://www.linkedin.com/in/daniel-ward-dev)
-  danielwarddev.bsky.social





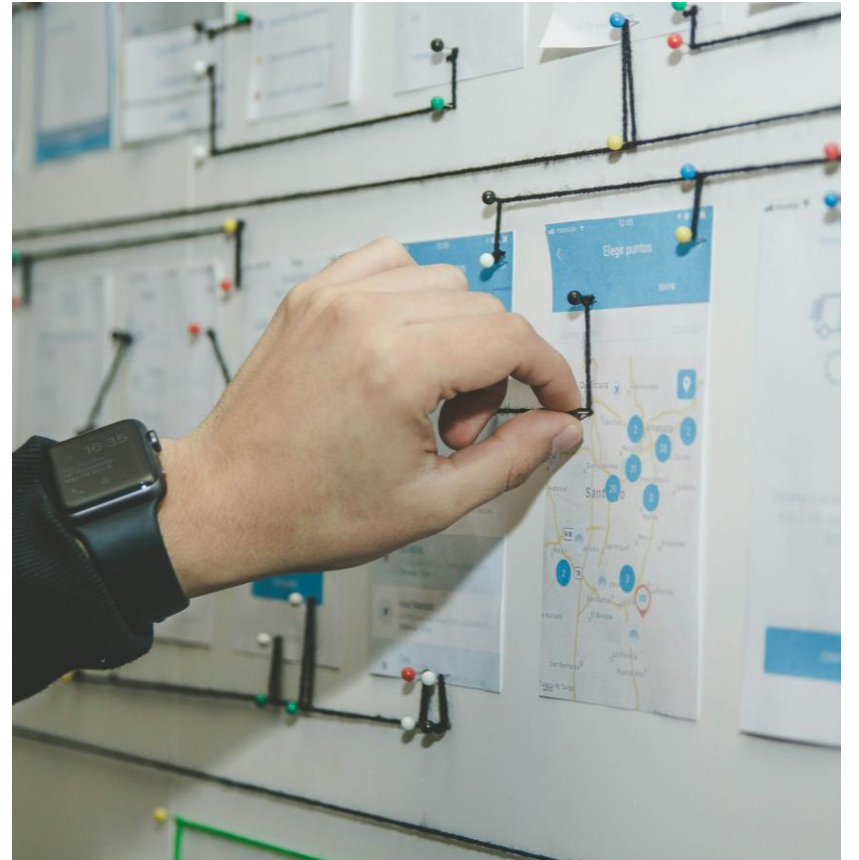
<https://www.meetup.com/sadnug/>

Outline

- Copilot features in both VS Code and CLI
- Good practices with using Copilot
- Create our own kanban app

What's a “real project?”

- Enterprise vs. startup
- Audience size
- Money size



Setting up

- Fork <https://github.com/danielwarddev/copilot-workshop-geeksand>
- Follow `workshop-content/00-setup.md`

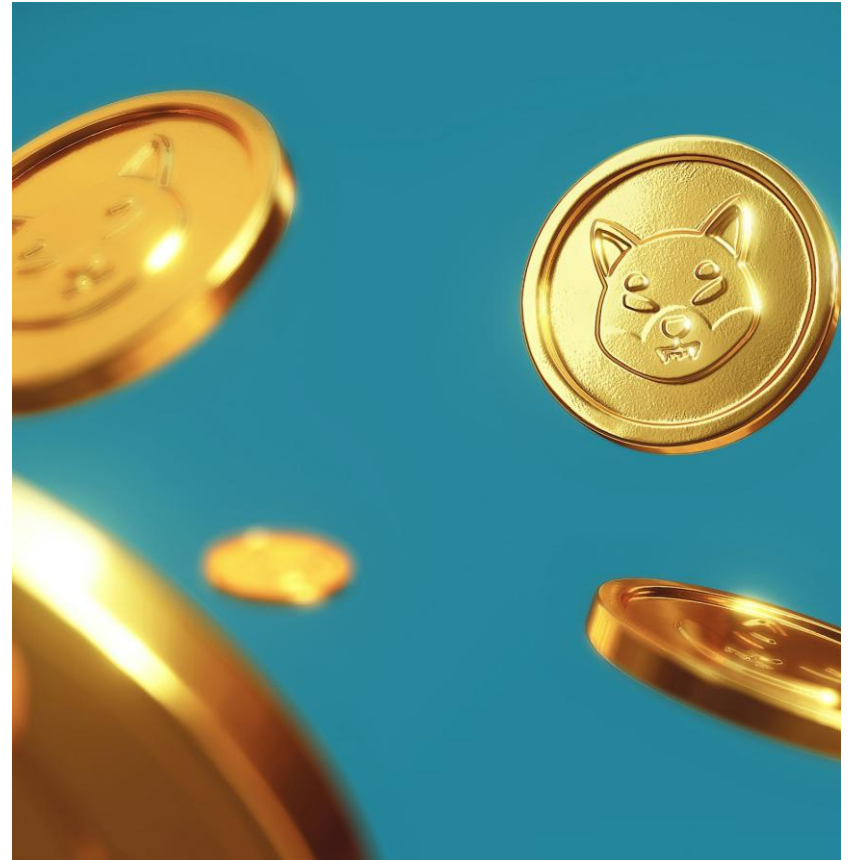
Copilot's value proposition

- Premium requests vs tokens
- Works with other conventions
 - AGENTS.md
 - CLAUDE.md
 - .claude/skills
- Use Claude/Codex harness
- Use any model family
- BYOK
- Offline with local models



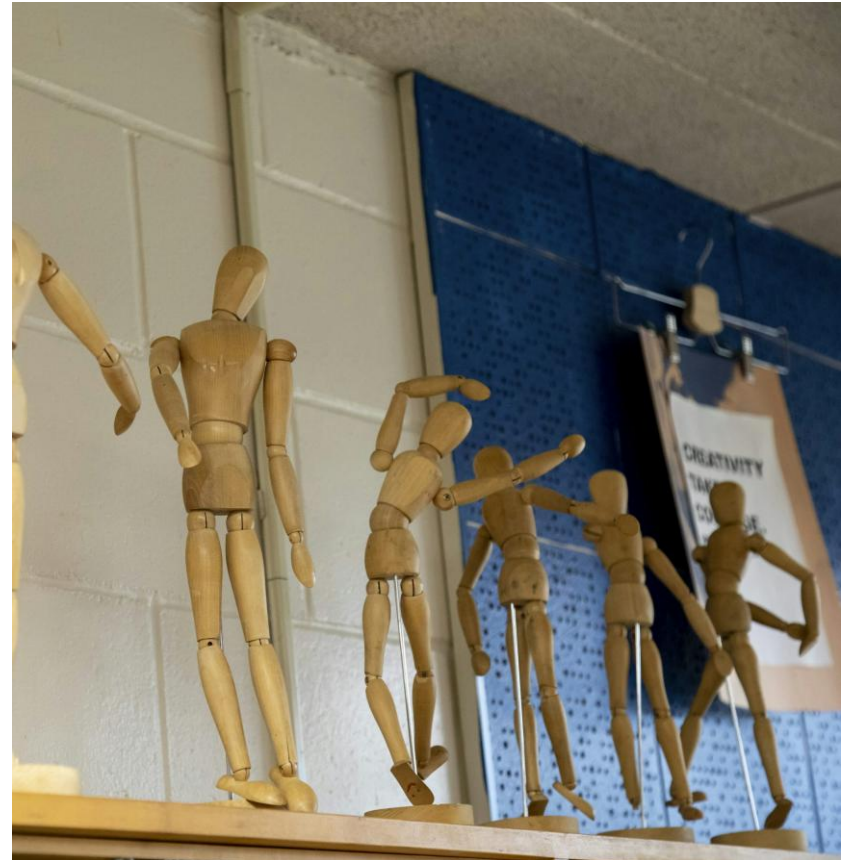
Premium requests vs tokens

- 1 message sent = 1 premium request
- 50 on the free plan
- Subagents don't consume an additional premium request



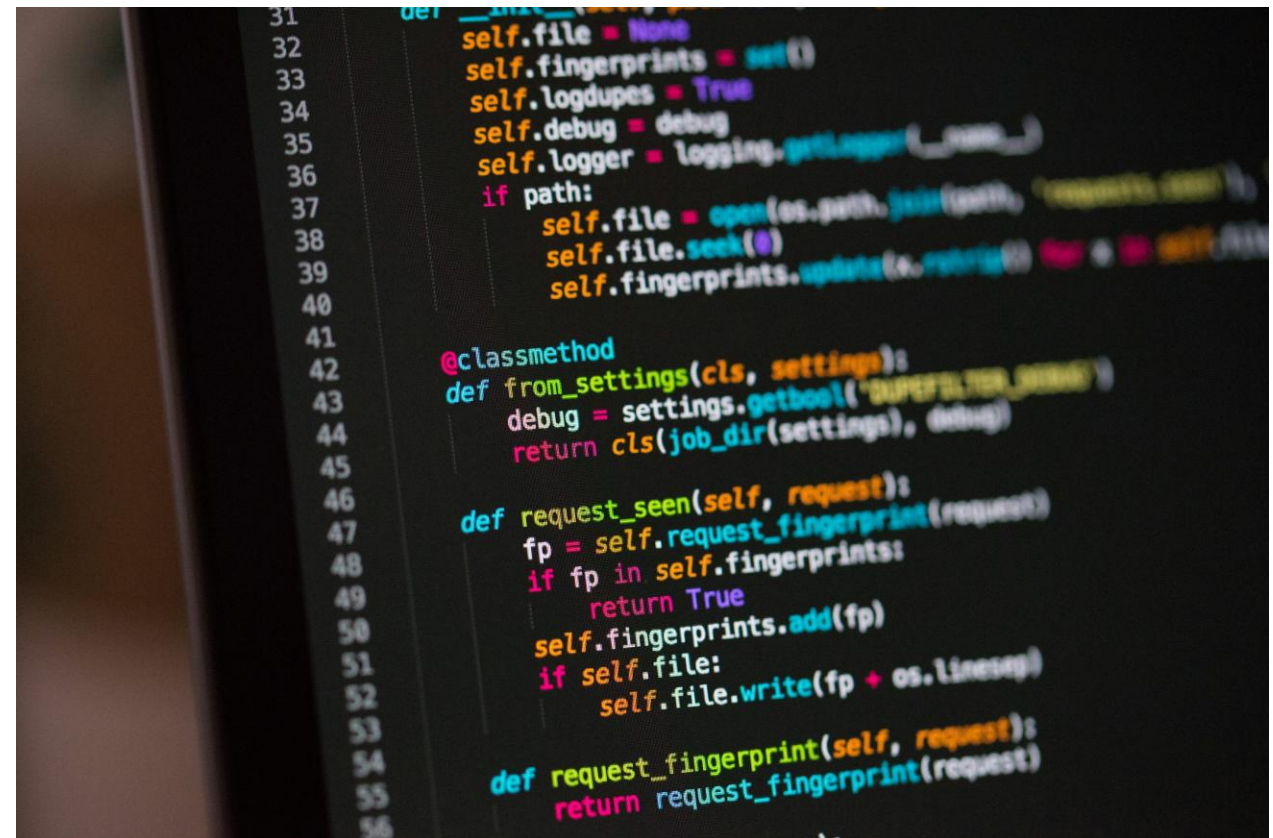
Free plan models

- Claude Haiku 4.5 (0.33x)
- GPT-4.1 (0x)
- GPT-5 mini (0x)
- Grok Code Fast 1 (0.25x)
- Raptor mini (0x)



VS Code vs CLI

- Different teams
- Some differences in features
- Use whichever you like better!



```
31 def __init__(self, path):
32     self.file = None
33     self.fingerprints = set()
34     self.logdupes = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, "requests.txt"),
39                         "a")
40         self.file.seek(0)
41         self.fingerprints.update(e.request() for e in self.requests)
42
43 @classmethod
44 def from_settings(cls, settings):
45     debug = settings.getbool("DEBUG")
46     return cls(job_dir(settings), debug)
47
48 def request_seen(self, request):
49     fp = self.request_fingerprint(request)
50     if fp in self.fingerprints:
51         return True
52     self.fingerprints.add(fp)
53     if self.file:
54         self.file.write(fp + os.linesep)
55
56 def request_fingerprint(self, request):
57     return request_fingerprint(request)
```

Ownership

- You still own the work
- Accepting AI's work implies you know what “correct” looks like

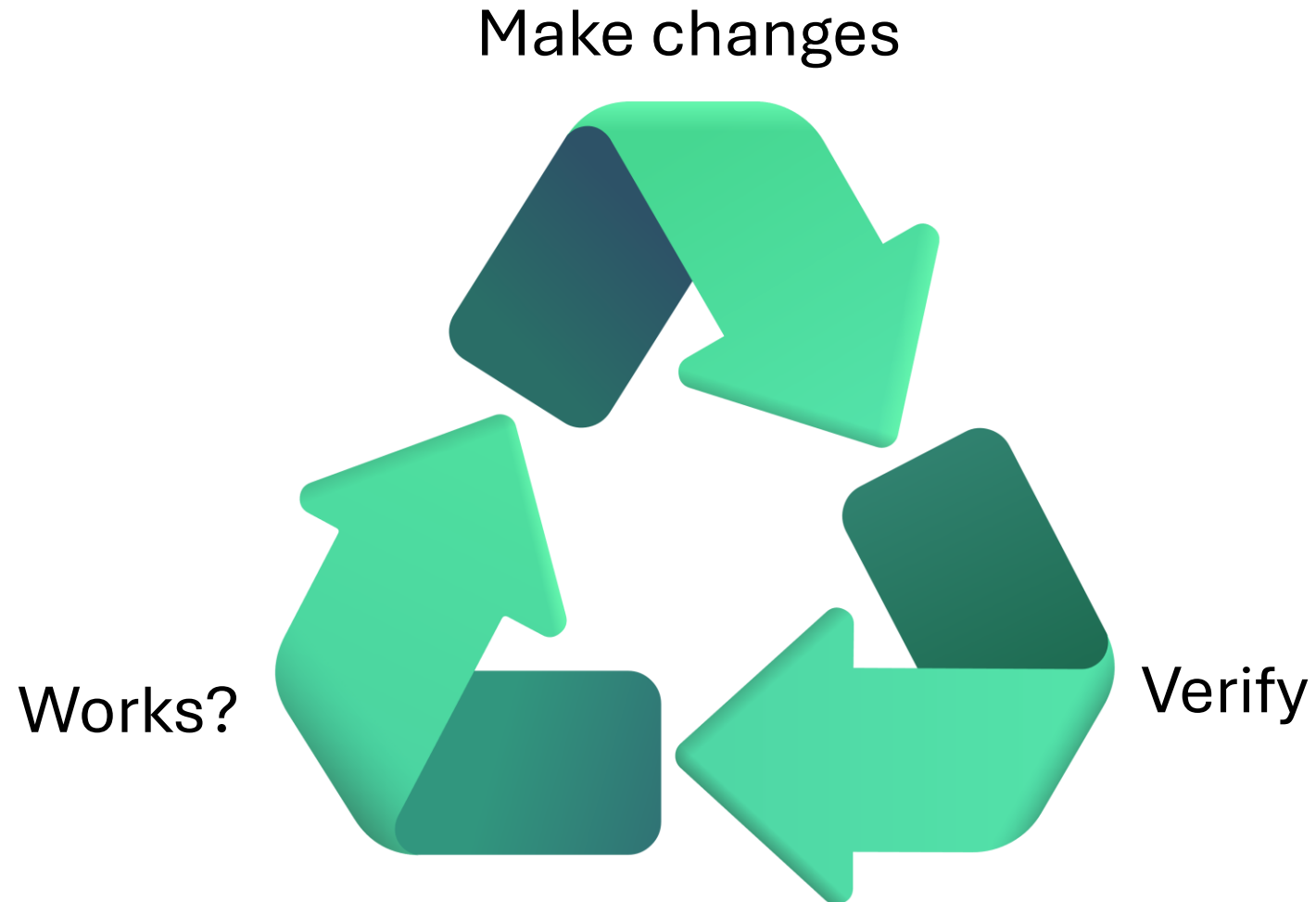


Litmus test

- Prod issues
- Work in small chunks so you can review

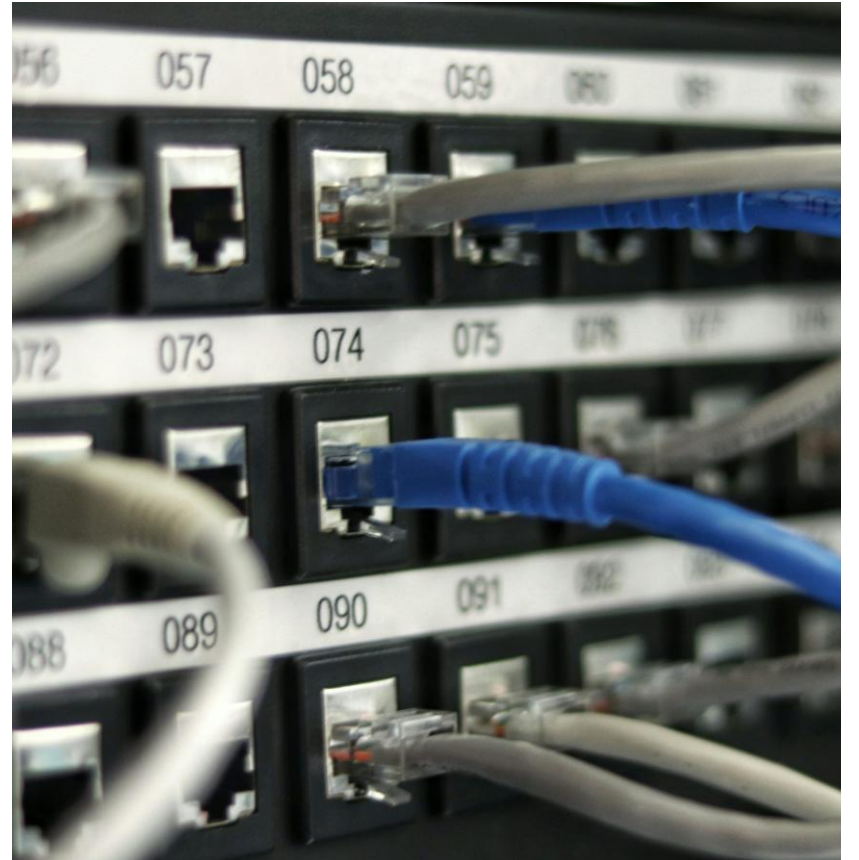


Feedback loops



MCP

- Model context protocol
- A **protocol** that allows us to give additional **context** to the **model** in a standardized way



Good practices

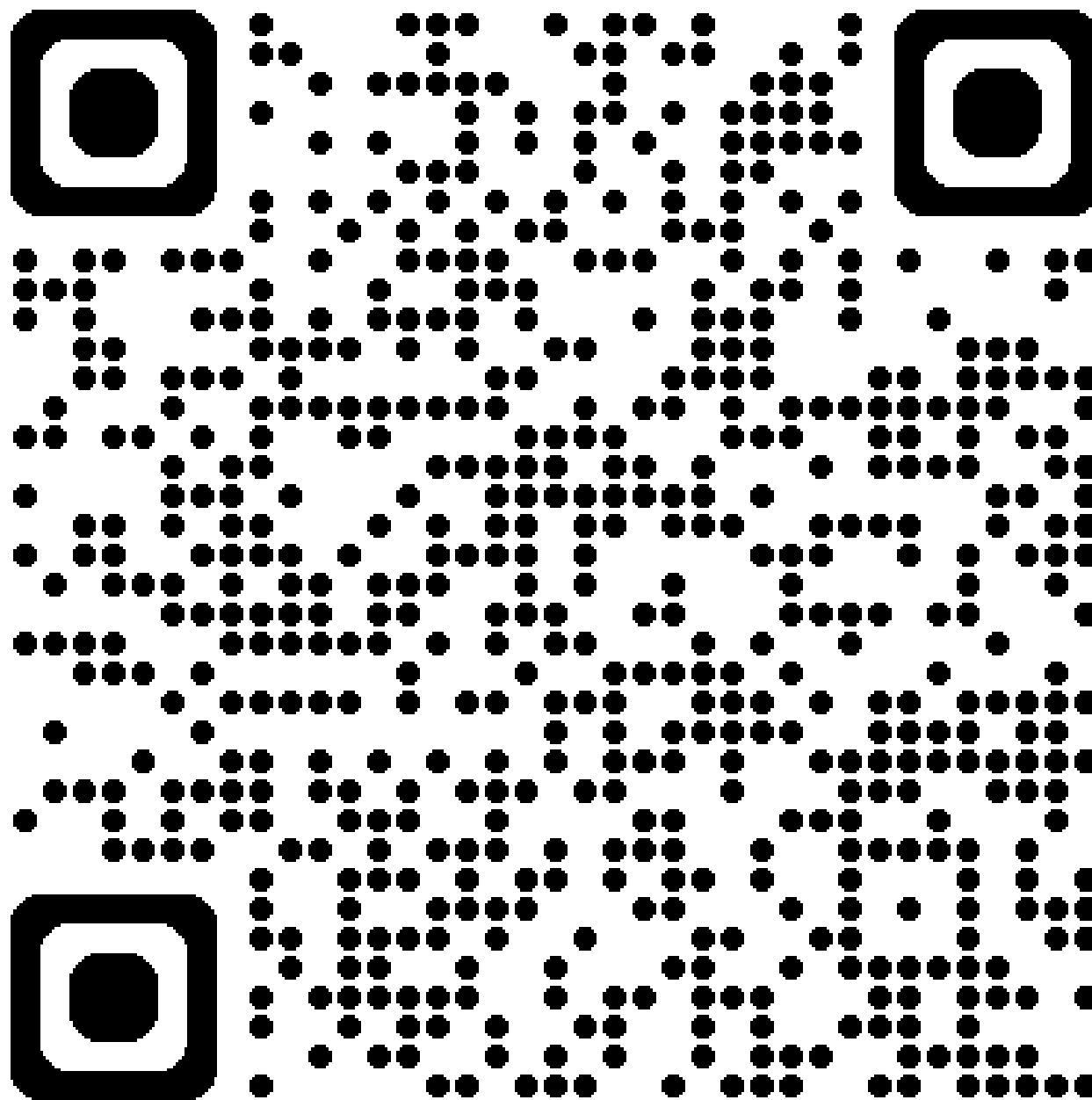
- Small chunks + review
- Feedback loop
- Build up instructions/skills over time
- AI doesn't replace learning new domains
 - This is how juniors can learn



Recap

- Copilot features
 - Instructions, modes, prompts, agents, skills, hooks, MCP, subagents, more
- Practices
 - Review output
 - Work in small chunks
 - Build up instructions over time
 - AI doesn't replace learning

Thank you!



Thank you!

