

AI-Driven Development: Best practices



Vladimir Kovtunovskiy

AI Product Engineer

I build AI-native products and automations

<https://www.linkedin.com/in/vvkai/>

Intro

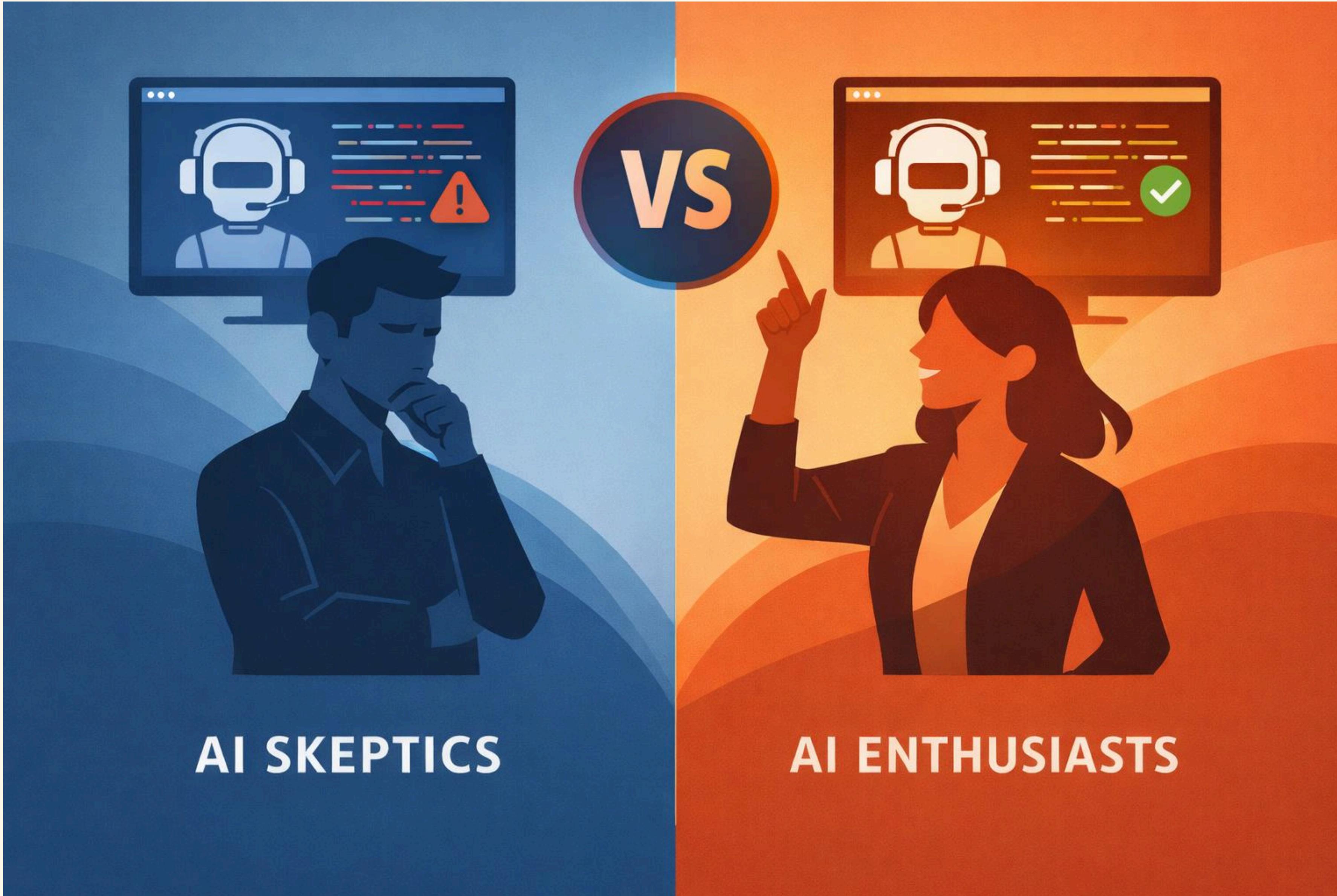
What is this talk about:

- AI Coding = Agentic Coding = AI-Driven Development
- but not Vibe Coding!
- Identify main mistakes and problems
- Provide a solution/approach that worked in real projects

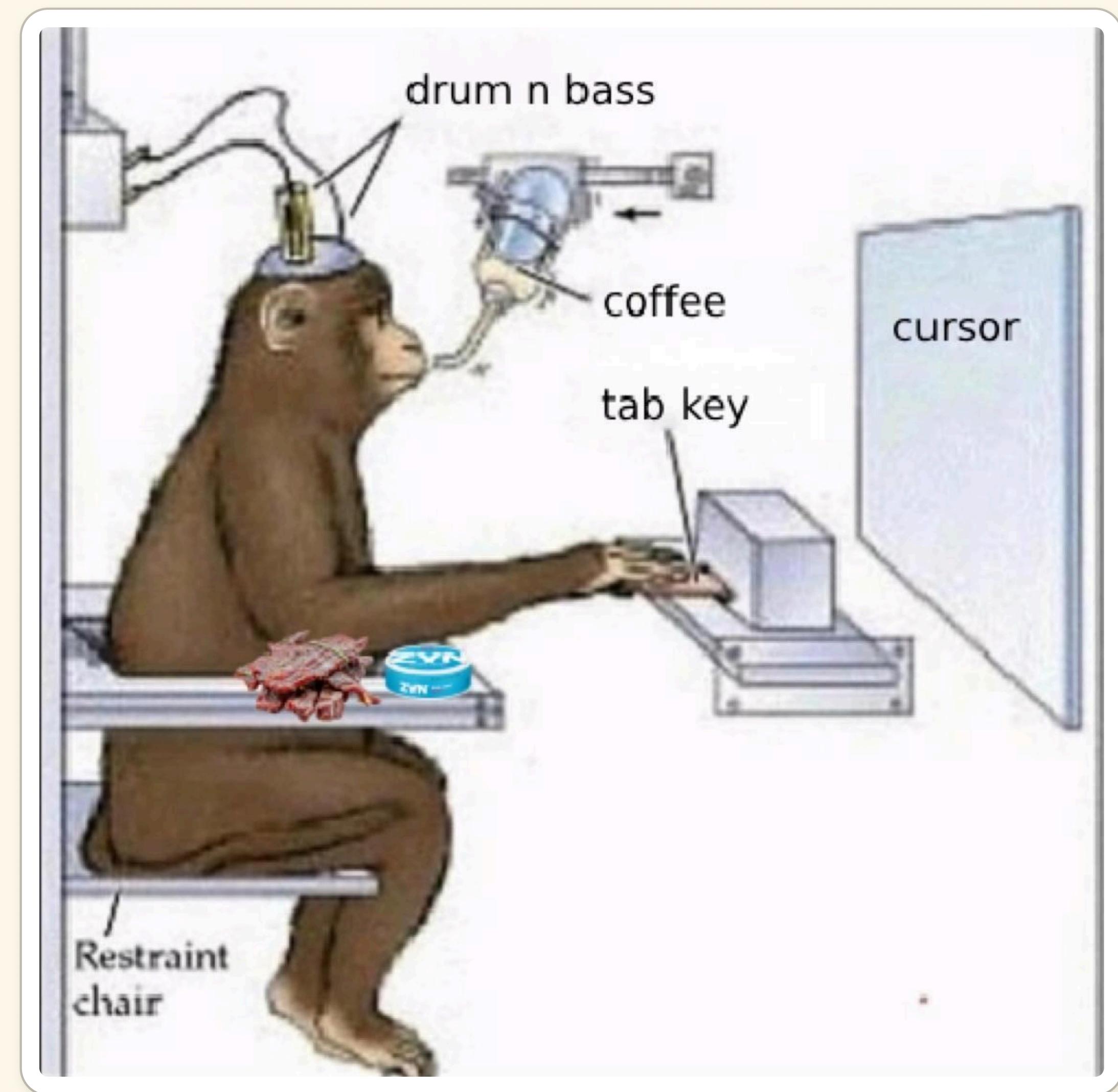
This talk is not about

- What's best coding Agent?
- best MCP to use?
- best LLM for Coding?

Skeptics vs Enthusiasts



Skeptics vs Enthusiasts



Skeptics vs Enthusiasts

December 12, 2025 Engineering

How we used Codex to build Sora for Android in 28 days

By Patrick Hum and RJ Marsan, Members of the Technical Staff

Satya Nadella says as much as 30% of Microsoft code is written by AI

Why such a big difference?

MISTAKE # 1: EXPECTATIONS

Mistake # 1: having the wrong expectations

1. New Projects vs Legacy Code
2. Python vs C++
3. 10k lines of code vs 10 million
4. Project for 1000 users vs 1b users

MISTAKE # 1: EXPECTATIONS

Solution #1: Set the correct Expectations for your project

1. Limitations of coding Agents
2. Type of work where they excel at
3. Don't expect one-shot miracles

MISTAKE # 2: MINDSET

Mistake # 2: keeping the old mindset/approach

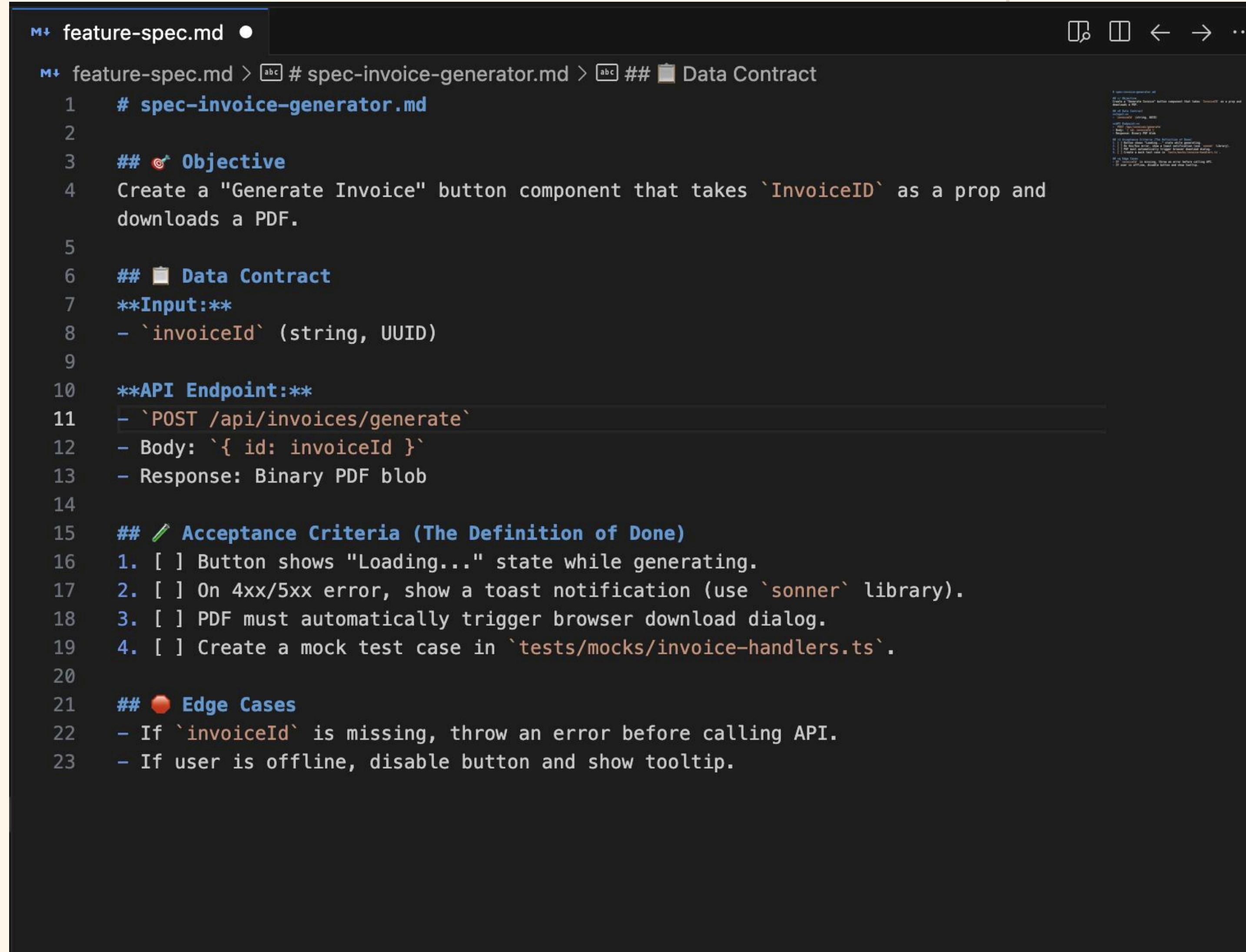
1. Thinking about code only
2. Assume Agent knows the same as you
3. Not transforming your role into part-time manager/architect/team-lead

MISTAKE # 2: MINDSET

Solution # 2: Spec-Driven Development

- Create artifacts that drive the entire development process
- Specification = single source of truth
- Intent is codified, not left to guesswork
- AI acts as literal-minded pair programmer, not guessing engine
- Supports iterative refinement and learning

SPEC-DRIVEN DEV CHECKLIST



```
feature-spec.md
feature-spec.md > # spec-invoice-generator.md > ## Data Contract
# spec-invoice-generator.md
## 🔎 Objective
Create a "Generate Invoice" button component that takes `InvoiceID` as a prop and
downloads a PDF.
## 📄 Data Contract
**Input:**
- `invoiceId` (string, UUID)
**API Endpoint:**
- `POST /api/invoices/generate`
- Body: `{ id: invoiceId }`
- Response: Binary PDF blob
## ✂️ Acceptance Criteria (The Definition of Done)
1. [ ] Button shows "Loading..." state while generating.
2. [ ] On 4xx/5xx error, show a toast notification (use `sonner` library).
3. [ ] PDF must automatically trigger browser download dialog.
4. [ ] Create a mock test case in `tests/mocks/invoice-handlers.ts`.
## 🚧 Edge Cases
- If `invoiceId` is missing, throw an error before calling API.
- If user is offline, disable button and show tooltip.
```

Spec-driven Check List

- ✓ Feature Description
[Objective] + [Context]
- ✓ User Flows
- ✓ Technical Constraints
- ✓ Edge Cases
- ✓ Acceptance Criteria

MISTAKE #3: NO RULES

Mistake #3: Not Preparing the Repo for Agent before coding

1. Agent won't know your preferences
2. Keep repeating the same mistakes
3. Might be dangerous

MISTAKE #3: NO RULES

Solution # 3:
Claude.md/Agents.md +
Docs

- Onboard the agent every session
- Add WHY / WHAT / HOW
- WHY: 1–2 lines on the project purpose
- WHAT: a short map of the repo
- HOW the *few* commands needed to validate changes (install, test, typecheck, lint, build, verify)

CLAUDE.MD EXAMPLE

```
m+ claude.md ●
m+ claude.md > # CLAUDE.md > ## How to Work on This Project (HOW)
1  # CLAUDE.md
2
3  ## Purpose (WHY)
4
5  This repository contains a production AI system designed to automate structured data
6  extraction and analysis from unstructured inputs.
7
8  ## System Overview (WHAT)
9
10 ## Tech Stack
11 - Python (core logic)
12 - FastAPI (API layer)
13
14 ## Repository Structure
15 - `/app` -- application entry points and API
16 - `/agents` -- LangGraph agent definitions
17
18
19 ## How to Work on This Project (HOW)
20
21 - Prefer small, incremental changes
22 - Maintain explicit reasoning steps in agent logic
23 - Never assume behavior--verify via tests or evaluation runs
24 - If uncertain, inspect existing patterns before introducing new ones
25 - Prioritize correctness and clarity over performance unless stated otherwise
26
27 When possible, validate changes using existing evaluation scripts.
28
29
30 ## Progressive Disclosure (Read When Needed)
31
32 Additional context is available in `/docs`:
33
34 - `architecture.md` -- system and data-flow overview
35 - `running_tests.md` -- how to run tests and evals
```

Claude.md rules:

- ✓ Be universally relevant
- ✓ Keep it short and high-signal
(50-70 lines ideally)
- ✓ Avoid instruction overload
- ✓ Optimize for verification, not style

Docs Checklist

- ✓ Architecture
- ✓ Tech Stack/Schema dbs
- ✓ How to run the project
- ✓ Key Concepts & Dev Conventions

Implementation Checklist: Before Coding

- ✓ Create concise CLAUDE.md (or AGENTS.md)
- ✓ Define WHAT, WHY, HOW for your project
- ✓ Set up spec-driven docs (db_schemas, architecture)
- ✓ Configure linters/formatters for code style
- ✓ Test agent behavior with clear test cases

Great, can we start writing a code now?

AGENDA

Mistake #4: Rushing to code before having a plan

1. You can miss relevant details
2. Agent can guess if context missing
3. Waste of time/tokens

AGENDA

Solution # 4: Plan & Act

- Task Description & Context
- Use Plan mode
- Discuss the plan with Agent
- Check the plan
- Implement it step by step

Now it's time to code... finally!

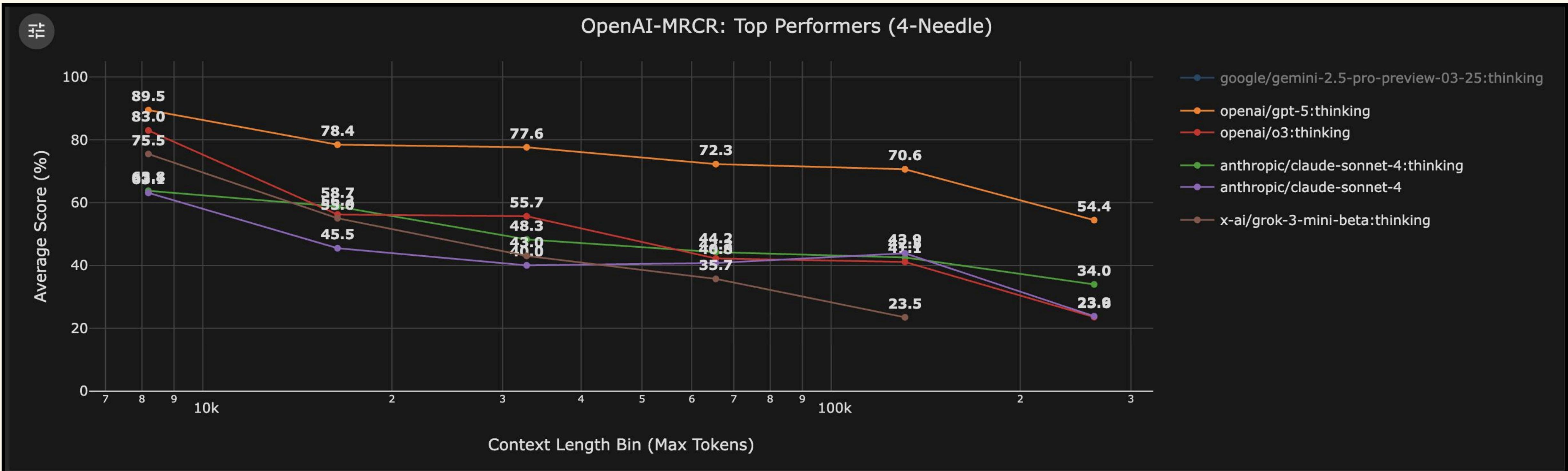
AGENDA

Mistake #5: not paying attention to your context

1. Agent forgets important rules
2. Context Clash
3. Context Distraction

Context Overview

ContextArena.ai



Context is a limited and precious resource.

AGENDA

Solution # 5: Context Managment

- New task = new chat
- Keep rules & docs up to date
- Always check what's inside your Agent context
- Split tasks into minimal steps

Context Management

REAL CASE

DB (wiki.js) integration into
existing custom CRM
(100k lines of code)

~45 minutes

Preparation time

~30 minutes

Coding Agent work time

~6 hours

Completion time without AI

Summary

- ✓ Set your expectations right
- ✓ Adopt a new spec-driven mindset
- ✓ Set up the rules and guidelines for the Agent
- ✓ Always Plan before acting
- ✓ Manage your Context

Any Questions?



Vladimir Kovtunovskiy

AI Product Engineer

I build AI-native products and automations

<https://www.linkedin.com/in/vvkai/>



Spec-Driven Development
check-list