KISSKI
KI-Servicezentrum für sensible und kritische Infrastrukturen

Sascha Safenreider

KI-Methoden im akademischen Alltag: Programmieren mit KI

Part 2

Outline

5 Fundamental Skills of Agentic Coding

- **Context:** Understanding the context and constraints of the project.
- **Thinking:** Understanding the problem and identifying the key elements.
- **Framework:** Choosing the right framework and tools for the project.
- **Checkpoints:** Setting milestones and tracking progress.
- **Debugging:** Identifying and fixing errors.

5 Fundamental Skills of Agentic Coding

5 Fundamental Skills of Agentic Coding: Context

Understanding the full picture before you start coding

- Context means knowing what you're building, for whom, and why
- Good context gathering prevents building the wrong thing
- Tools: Mind maps and Product Requirements Documents (PRD)
- The clearer your context, the better your AI results

5 Fundamental Skills of Agentic Coding: Thinking

Use Case: PRD Step 1 - Logical Thinking

Project Overview (Logical Thinking)

- The goal of this project is to develop an AI-powered system that creates personalized weekly meal plans for casual households
- The system will take into account factors such as work schedule, cooking skills, dietary restrictions, food preferences, and grocery budget
- It will also consider available kitchen equipment and pantry inventory to ensure the meal plan is practical and efficient
- For users with limited cooking experience or literacy, the plan will include visual recipe cards to make preparation easier

5 Fundamental Skills of Agentic Coding: Thinking

Use Case: PRD Step 2 - Analytical Thinking

Skills Required (Analytical Thinking)

- Examples:
  - My programming language (e.g. Python)
  - Recipe and Nutrition Data Processing
  - API Key to a certain AI model
  - Image Processing (for visual recipe cards)
  - UI development
    - Maybe in addition: More specific Frontend/Backend stack

5 Fundamental Skills of Agentic Coding: Thinking

Use Case: PRD Step 3 - Computational Thinking

Key Features (Computational Thinking)

- **Milestone 1:** Personalized Meal Plan Engine
  - AI-Based Meal Plan Generation:
    - Generate weekly meal plans based on user preferences, dietary needs, available cooking time, and budget
    - Suggest ingredient substitutions based on pantry inventory and local grocery availability
- **Milestone 2:** Contextual Customization:
  - Adapt instructions based on user cooking skill level and literacy—provide visual recipe cards for beginners or busy users who prefer quick visual scanning
  - Recommend batch cooking or quick-prep options for users with especially busy weeks

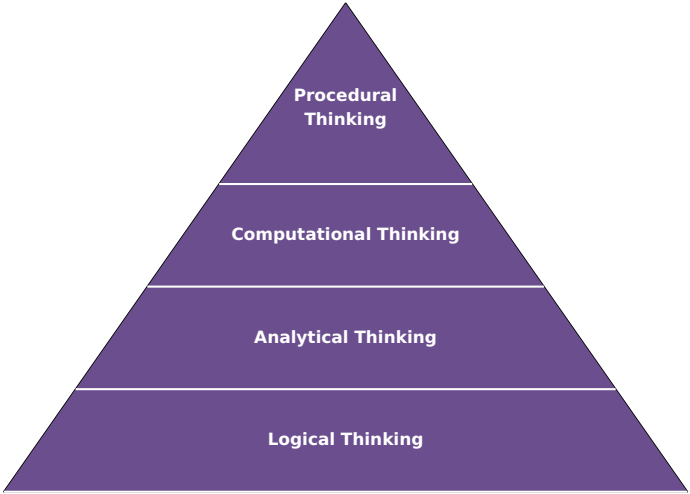
5 Fundamental Skills of Agentic Coding: Thinking

Use Case: PRD Step 4 - Procedural Thinking

Detailed Context (Procedural Thinking)

- User lifestyle data: work schedule, cooking skills, dietary restrictions, preferences, grocery budget, available equipment
- **Client Information:**
  - The client is a lifestyle-focused health coach based in Germany
  - They work primarily with urban professionals in high-paced careers who want to maintain a healthy diet without excessive time spent on meal planning or cooking
  - The client's focus is on improving daily nutrition habits through simple, personalized, and practical solutions that fit real-world schedules

5 Fundamental Skills of Agentic Coding: Thinking



5 Fundamental Skills of Agentic Coding: Framework

Guide the AI to the right tools

Why?

If you clearly tell the AI *what* tools to use, it can give you better fitting examples and instructions.

- **What am I building?** (e.g., a simple web app)
- **Which tools?** (e.g., React, HTML/CSS/JavaScript)
- Optional: **Any extras?** (e.g., animations, special styling)

Example prompt

```
I want to create a small web app with React.
Use HTML, CSS, and JavaScript as the base, plus Tailwind CSS for styling.
Start with a homepage that shows a title and a "Start" button.
```

5 Fundamental Skills of Agentic Coding: Checkpoints

Why checkpoints and version control matter

Key idea

Things will break — and that’s normal. Checkpoints let you save working versions of your project so you can always go back.

- You can save your project’s current state
- If today’s changes break things, you can restore yesterday’s version
- Helps you try new ideas without fear

5 Fundamental Skills of Agentic Coding: Debugging

Finding and fixing problems

Key idea

🔍 Debugging is about finding out what's wrong and fixing it step by step.

- Identify:
  - 🔍 Where the problem is
  - ⚠️ What the problem is
- Try different solutions until it works 🔧

Example prompt

```
My app crashes when I click the login button.
Help me find where the error is and fix it.
```

5 Fundamental Skills of Agentic Coding

Outline

5 Fundamental Skills of Agentic Coding: Context

Now we are ready to merge our PRD Steps into one document:

- **Project Overview (Logical Thinking)**
  - The goal of this project is to develop an AI-powered system that creates personalized weekly meal plans for casual households
  - The system will take into account factors such as work schedule, cooking skills, dietary restrictions, food preferences, and grocery budget
  - It will also consider available kitchen equipment and pantry inventory to ensure the meal plan is practical and efficient
  - For users with limited cooking experience or literacy, the plan will include visual recipe cards to make preparation easier
- **Skills Required (Analytical Thinking)**
  - Python
  - Recipe and Nutrition Data Processing
  - API Key
  - Image Processing (for visual recipe cards)
  - UI development (optionally: more specific Frontend/Backend stack)
- **Key Features (Computational Thinking)**
  - **Milestone 1:** Personalized Meal Plan Engine
    - Generate weekly meal plans based on user preferences, dietary needs, available cooking time, and budget
    - Suggest ingredient substitutions based on pantry inventory and local grocery availability
  - **Milestone 2:** Contextual Customization
    - Adapt instructions based on user cooking skill level and literacy
    - Provide visual recipe cards for beginners or busy users who prefer quick visual scanning
    - Recommend batch cooking or quick-prep options for users with especially busy weeks
- **Detailed Context (Procedural Thinking)**
  - User lifestyle data: work schedule, cooking skills, dietary restrictions, preferences, grocery budget, available equipment
  - **Client Information:**
    - The client is a lifestyle-focused health coach based in Germany
    - They work primarily with urban professionals in high-paced careers who want to maintain a healthy diet without excessive time spent on meal planning or cooking
    - The client's focus is on improving daily nutrition habits through simple, personalized, and practical solutions that fit real-world schedules.

5 Fundamental Skills of Agentic Coding: Context

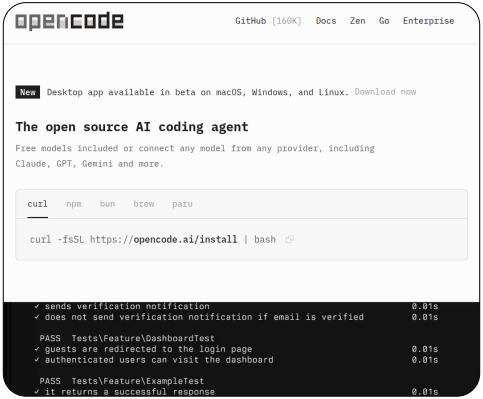
How to prompt for PRD

Help me make a PRD for an MVP app I'm looking for Agentic code.
I'm looking for an AI-powered app that creates personalized weekly meal plans for casual households. The system should consider work schedules, cooking skills, dietary restrictions, food preferences, and grocery budget. It should also account for available kitchen equipment and pantry inventory to ensure the meal plan is practical and efficient.
The app should provide visual recipe cards and step-by-step guides for users with limited cooking experience or literacy. It should suggest ingredient substitutions based on pantry items and local grocery availability.
Help me think through the rest of the app. For example: What are the core features? How will users interact with the app daily? How will the app adapt to changing user preferences or schedules? How will we handle accessibility for low-literacy users? How will socioeconomic factors influence recommendations?
We should think through the following questions: What is the product? How do users interact with it? What is the underlying logic behind it? How can we make it engaging and easy to follow?
I've also attached an example PRD that is well fleshed out as guidance. [Attach an example PRD]

Before You Start

You need two things before opening opencode:

1. **Opencode** installed on your machine
   Download from `https://opencode.ai`
2. **Your SAIA API key**
   Get this from KISSKI :
   `https://s.gwdg.de/QJUDQf`



Step 1: Set Your SAIA API Key

Linux / macOS:

```bash
echo 'export SAIA_API_KEY="your_api_key_here"' >> ~/.bashrc source ~/.bashrc
```

For zsh (common on newer Macs) replace `.bashrc` with `.zshrc`

Windows (PowerShell):

```PowerShell
[Environment]::SetEnvironmentVariable("SAIA_API_KEY", "your_api_key_here", "User")
```

Step 2: Install the SAIA Plugin

Linux / macOS:

bash

```
curl -fsSL https://raw.githubusercontent.com/jaisonlewis/opencode-saia-plugin/master/install.sh | bash
```

Windows (PowerShell):

PowerShell

```
Invoke-Expression (Invoke-WebRequest -Uri "https://raw.githubusercontent.com/jaisonlewis/opencode-saia-plugin/master/install.ps1" -UseBasicParsing).Content
```

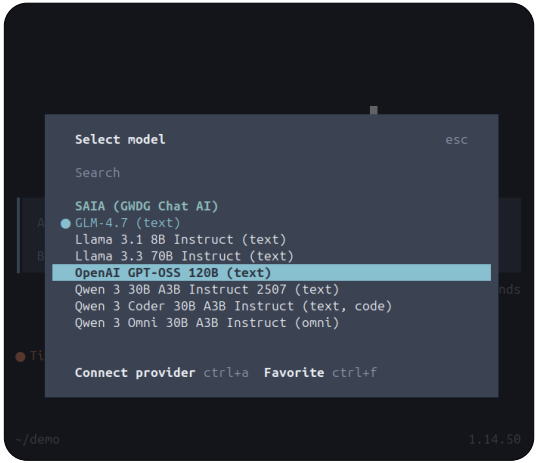
The script handles everything — plugin placement and registration.

Step 3: Launch and Verify

bash

```
opencode
```

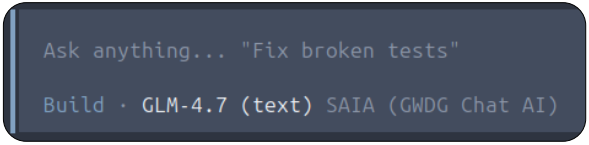
- First launch refreshes your SAIA model list in the background
- Type `/models` to confirm SAIA models appear
- If models are listed, everything is connected and working



Tour of the opencode Interface

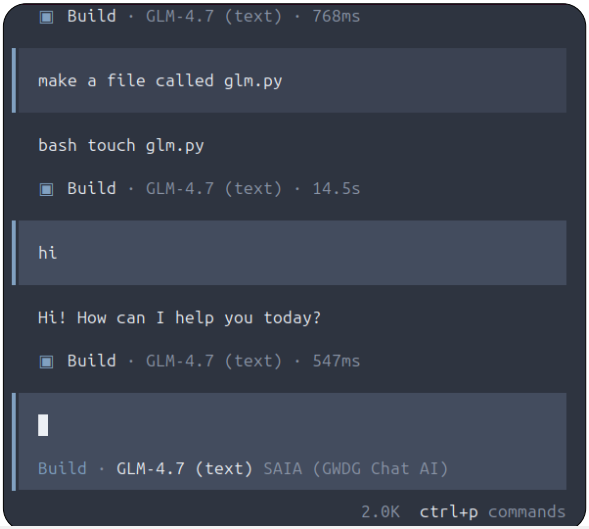
Tour of the opencode Interface

1. **Input bar** at the bottom — type prompts here



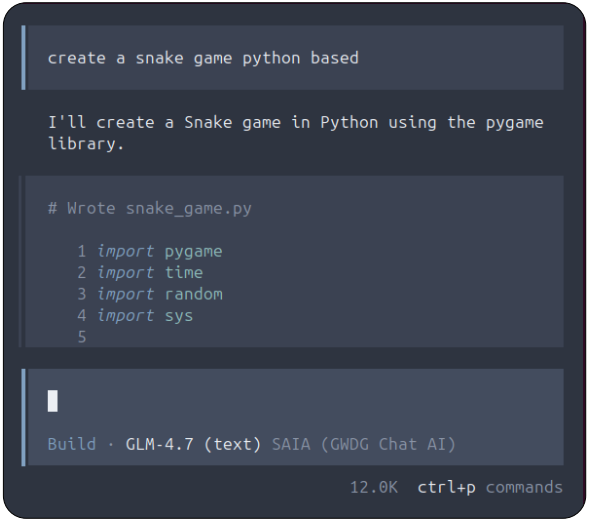
Tour of the opencode Interface

1. **Input bar** at the bottom — type prompts here
2. **Conversation area** — full session history



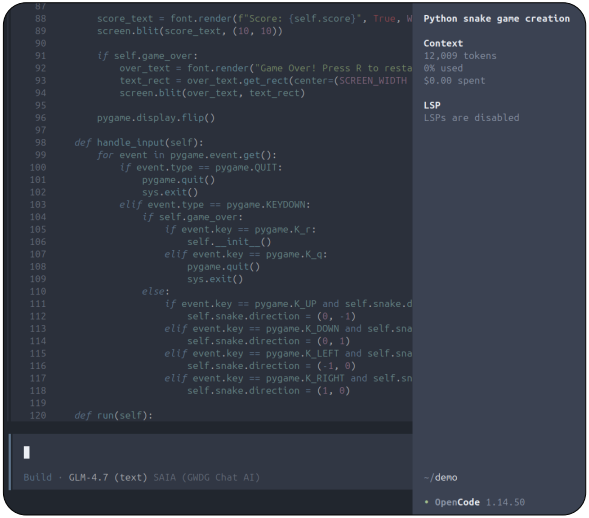
Tour of the opencode Interface

1. **Input bar** at the bottom — type prompts here
2. **Conversation area** — full session history
3. **AI actions in real time** — see files being read and edited



Tour of the opencode Interface

1. **Input bar** at the bottom — type prompts here
2. **Conversation area** — full session history
3. **AI actions in real time** — see files being read and edited
4. **Toggle Sidebar** — for more information



Tour of the opencode Interface

1. **Input bar** at the bottom — type prompts here
2. **Conversation area** — full session history
3. **AI actions in real time** — see files being read and edited
4. **Toggle Sidebar** — for more information

Intentionally simple — your focus is on building, not on the tool.

Exercise 1: Getting to Know opencode

MyBoard: First Contact

1. Open your project folder in the terminal
2. Launch opencode with `opencode`
3. Switch to `glm-4.7` using `/models`
4. Describe MyBoard in plain language and ask the AI to tell you what it understood
5. Practice `/undo`, `/models`, and `/compact`
6. **No files created yet**

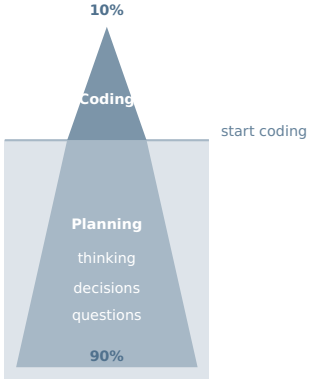
Tip: If the AI misunderstands your description, refine it and try again. This is normal.

Not All Models Are Equal

- Some models are built for **speed** — excellent at writing code
- Others are built for **deep reasoning** — better at thinking through problems
- Using a coding model to plan is like asking a builder to design your house
- They will start laying bricks immediately
- You only find out the rooms are in the wrong place once the walls are up

Front-loading Intelligence

- The more thinking you do **before** the first file is created, the less you undo later
- **Changing a plan costs nothing**
- Changing code that other code already depends on costs a lot
- This is what developers mean by *refactoring is expensive*
- A planning conversation that takes 20 minutes can save hours of rebuilding later



AGENTS.md and Context Files

- opencode reads `AGENTS.md` **automatically** at the start of every session
- Write rules once — the AI follows them throughout the whole project
- Lock in:
  - Naming conventions
  - Folder rules
  - Tech stack decisions
  - Always read SPEC.md instruction
- Use `/init` to create it — guided wizard

AGENTS.md

```
# project rules
- kebab-case file names
- components in src/components/
- plain HTML / CSS / JS
- always read SPEC.md
```

auto-read at session start

Outline

Agentic Coding Tools Overview

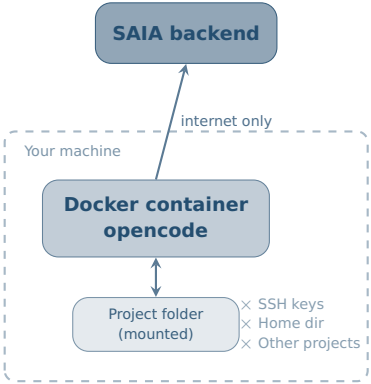
Which tool will your group test?

OpenCode

- Windsurf
- emergent.sh
- Lovable
- Cursor
- manus.im
- Base44

Running opencode Securely with Docker

- Agentic tools have **direct file system and command access**
- Simplest reliable containment: run Opencode inside a **Docker container**
- The container only sees the **folder you give it**
- It **cannot** reach your home directory, other projects or SSH keys
- Outbound internet only — enough to reach SAIA, nothing on your local network



Risks of Agentic Coding on a local machine

- Agents can overwrite or delete files and directories
- Local agents may access sensitive data (API keys, SSH keys, ...)
- Agents can trigger infinite loops (causes excessive hardware consumption)

Hands-on: Create an Agentic Coding Project

Plenary Task 30 Min

Please take your time to create an Agentic Coding project.
Consider the following:

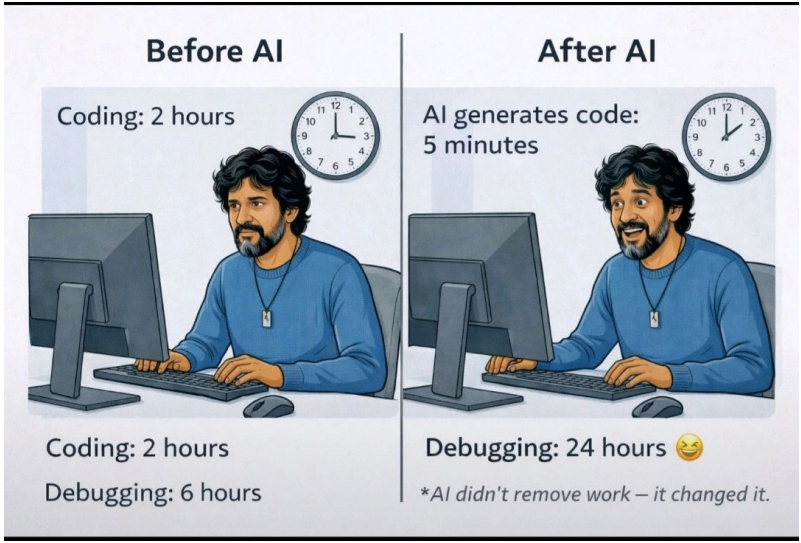
- PRD
- AGENTS.md
- Debugging
- Add features Step by Step

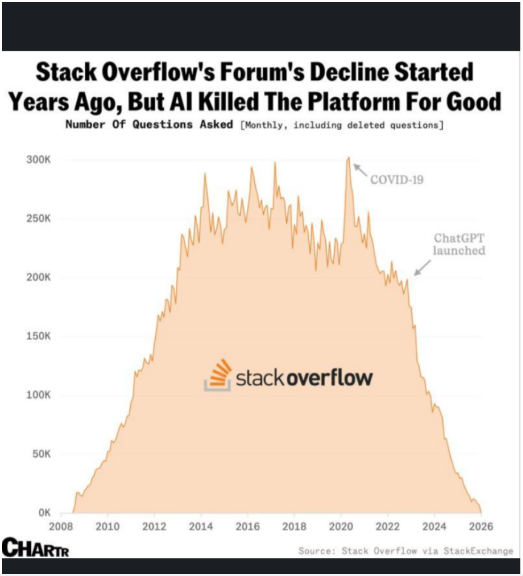
Interactive: Agentic Coding Discussion

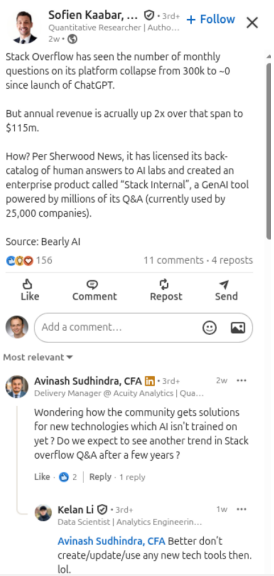
Plenary Task 10 Min

Please discuss the following questions in groups:

- What are the benefits and challenges of using Agentic Coding?
- How can Agentic Coding be applied in different industries and domains?
- What are the key skills and qualities required for successful Agentic Coding?


Before AI
Coding: 2 hours
Coding: 2 hours
Debugging: 6 hours
After AI
AI generates code:
5 minutes
Debugging: 24 hours
*AI didn't remove work – it changed it.


Stack Overflow's Forum's Decline Started Years Ago, But AI Killed The Platform For Good
Number Of Questions Asked [Monthly, including deleted questions]
300K
250K
200K
150K
100K
50K
0K
2008
2010
2012
2014
2016
2018
2020
2022
2024
2026
COVID-19
ChatGPT launched
stackoverflow
CHARTR
Source: Stack Overflow via StackExchange


Sofien Kaabar, ... • 3rd+
Quantitative Researcher | Autho...
2w •
+ Follow
Stack Overflow has seen the number of monthly questions on its platform collapse from 300k to ~0 since launch of ChatGPT.
But annual revenue is acrually up 2x over that span to $115m.
How? Per Sherwood News, it has licensed its back-catalog of human answers to AI labs and created an enterprise product called "Stack Internal", a GenAI tool powered by millions of its Q&A (currently used by 25,000 companies).
Source: Bearly AI
156
11 comments • 4 reposts
Like
Comment
Repost
Send
Add a comment...
Most relevant
Avinash Sudhindra, CFA • 3rd+
Delivery Manager @ Acuity Analytics | Qua...
2w
Wondering how the community gets solutions for new technologies which AI isn't trained on yet ? Do we expect to see another trend in Stack overflow Q&A after a few years ?
Like · 2 | Reply · 1 reply
Kelan Li • 3rd+
Data Scientist | Analytics Engineerin...
1w
Avinash Sudhindra, CFA Better don't create/update/use any new tech tools then. lol.

Outline

Conclusion

- We explored the idea of **Agentic Coding** as a new creative coding paradigm
- Learned the **5 Fundamental Skills**: Context, Thinking, Framework, Checkpoints, Debugging
- Practiced creating **PRDs** and discussed their importance
- Tested and evaluated different **AI coding tools** in practice
- Shared insights, benefits, and risks of relying on AI for coding

Take Home Message

Clear Vision → Clear PRD → Better AI Results

Agentic Coding empowers creativity and lowers technical barriers

But: Human review and security remain essential