

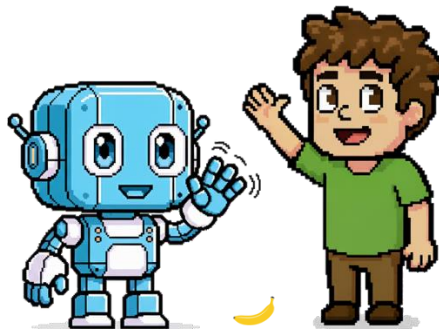
ML 04

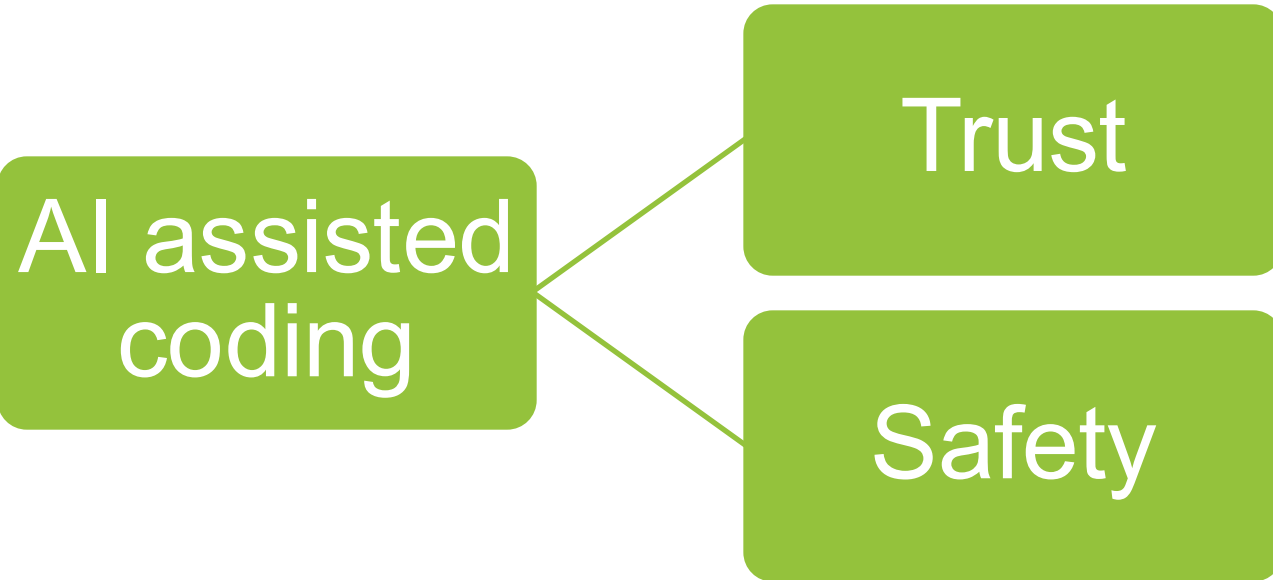
Neither Magic Wand Nor Terminator:

Finding the Sweet Spot in AI-Assisted Coding

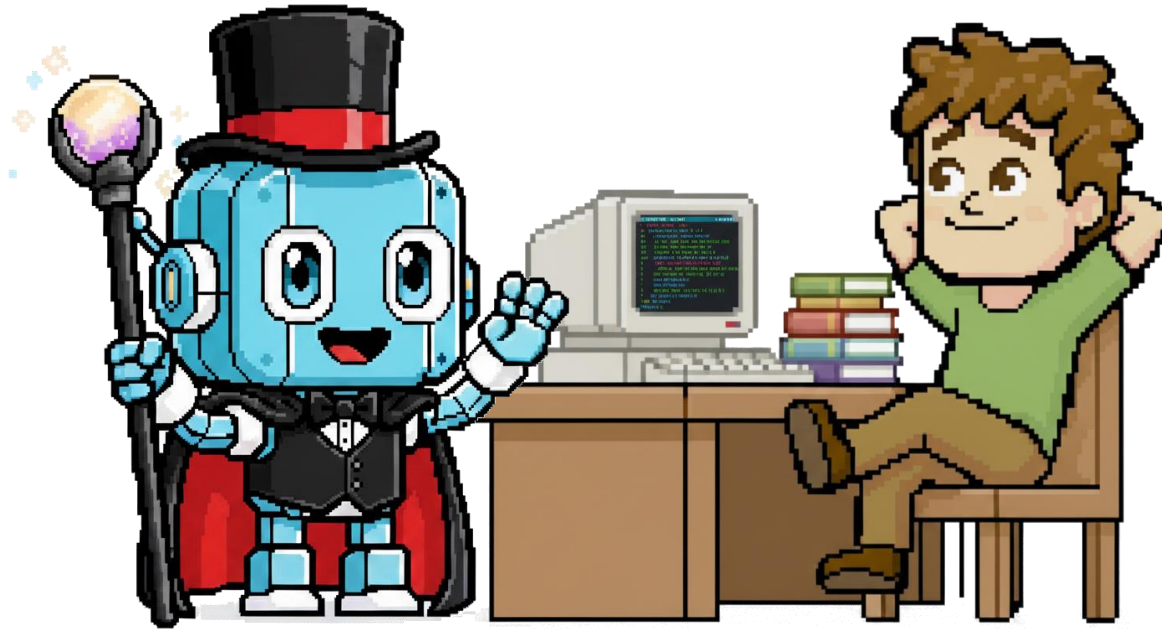
Hamburg – PHUSE EU Connect 2025

Josua Böser – josua.boeser@chrestos.de



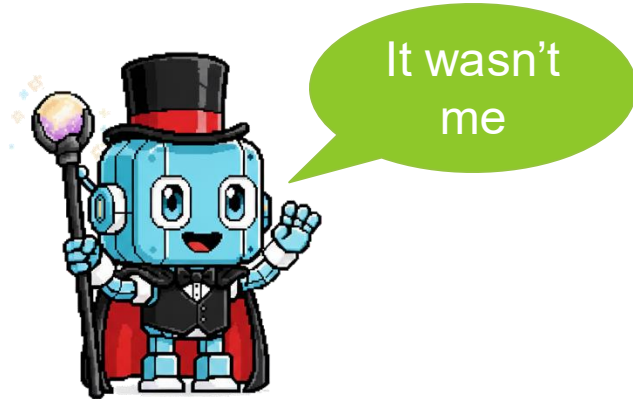


How developers would like to use AI



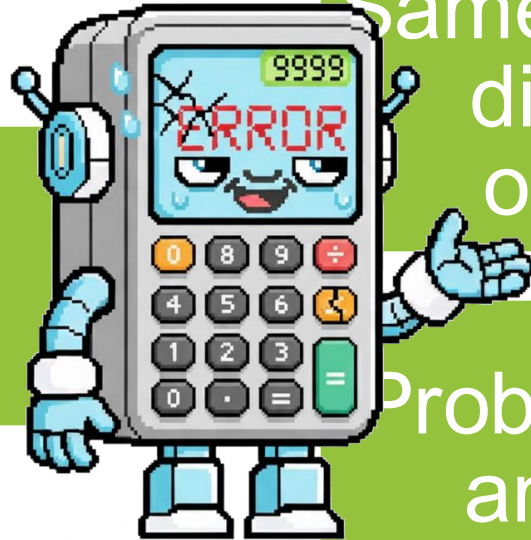
But AI make mistakes

- [Vibe coding service Replit deletes production database](#)
- [AI managing snack vending machine starts selling metal cubes](#)



And thinks differently

Probabilistic



Same question
different
outputs

Probable right
answers

No wizard but ...?



But humans make mistakes

- The 2000 problem - A \$548 Billion Software Crisis
- Mariner 1 - A missing hyphen causes \$18.5 Million damages
- Junior developer destroys production on day one
- When NASA lost a mars climate orbiter - a unit conversion error



No terminator but ... ?

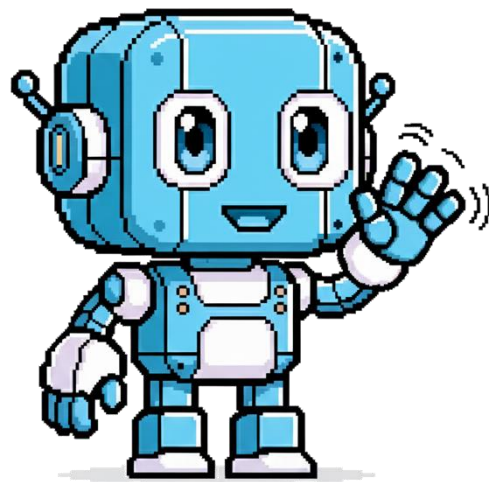
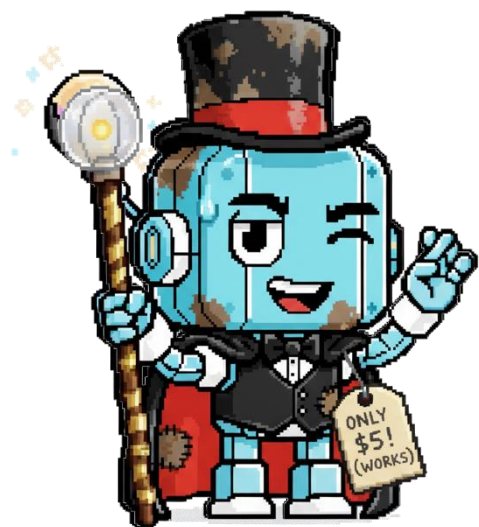


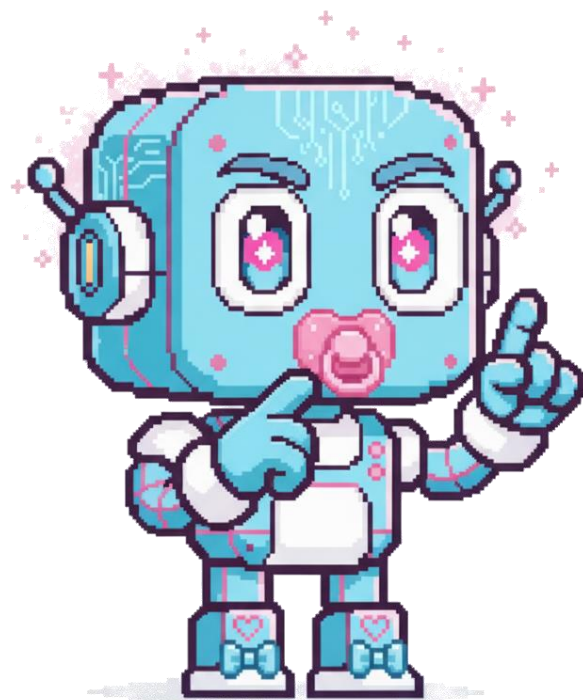
Impact of Early-2025 AI on Experienced Open-Source Developer

-19% Productivity

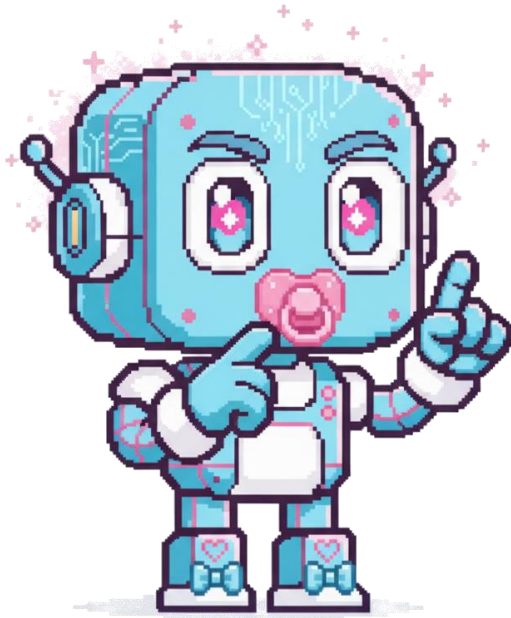


Who is AI?





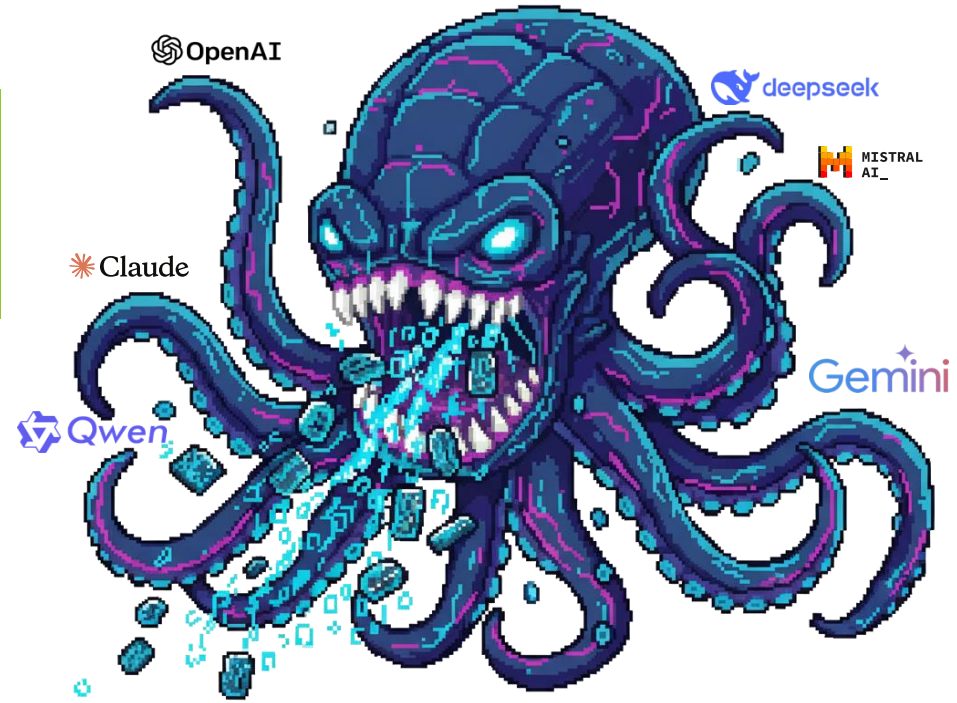
How to give a junior serious tasks?



- One task at a time
- Explain in detail
- Give examples
- Short without distractions
- No complex task
- Restrict the abilities
- Check every output

**What about my data when
I use AI?**

AI used “FEED”



It was super effective

What gives me more data security?

Online

Offline



Lumo



Ollama

**How can we use a junior AI
in an example?**

What is our setup



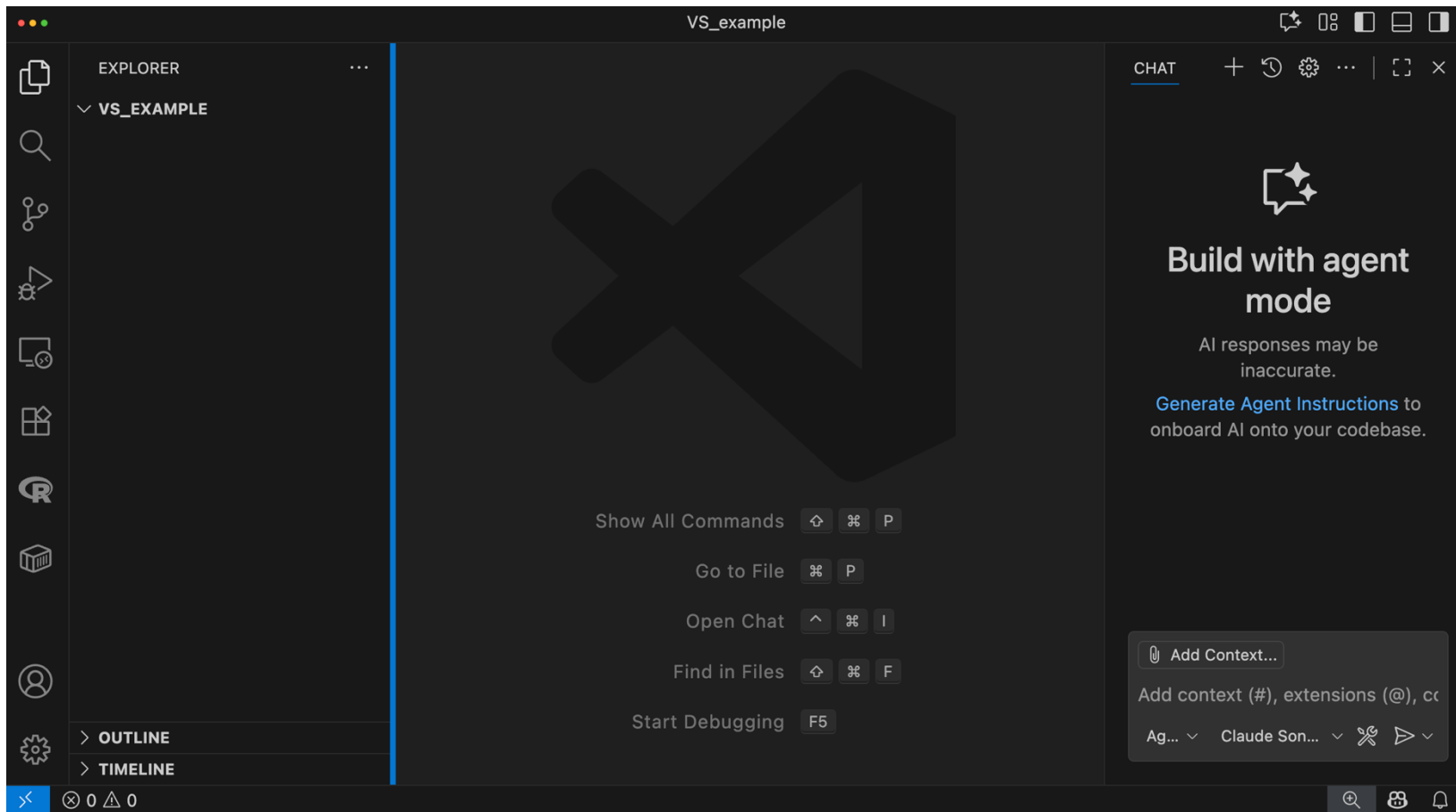
GitHub Copilot



git

Claude Sonnet 4.5





Case example: Learning a new language



Let us create a shiny app in R

Prompt:

1. Initialize git
2. Create a virtual environment (renv in R)
3. Generate the `.github/copilot-instructions.md`

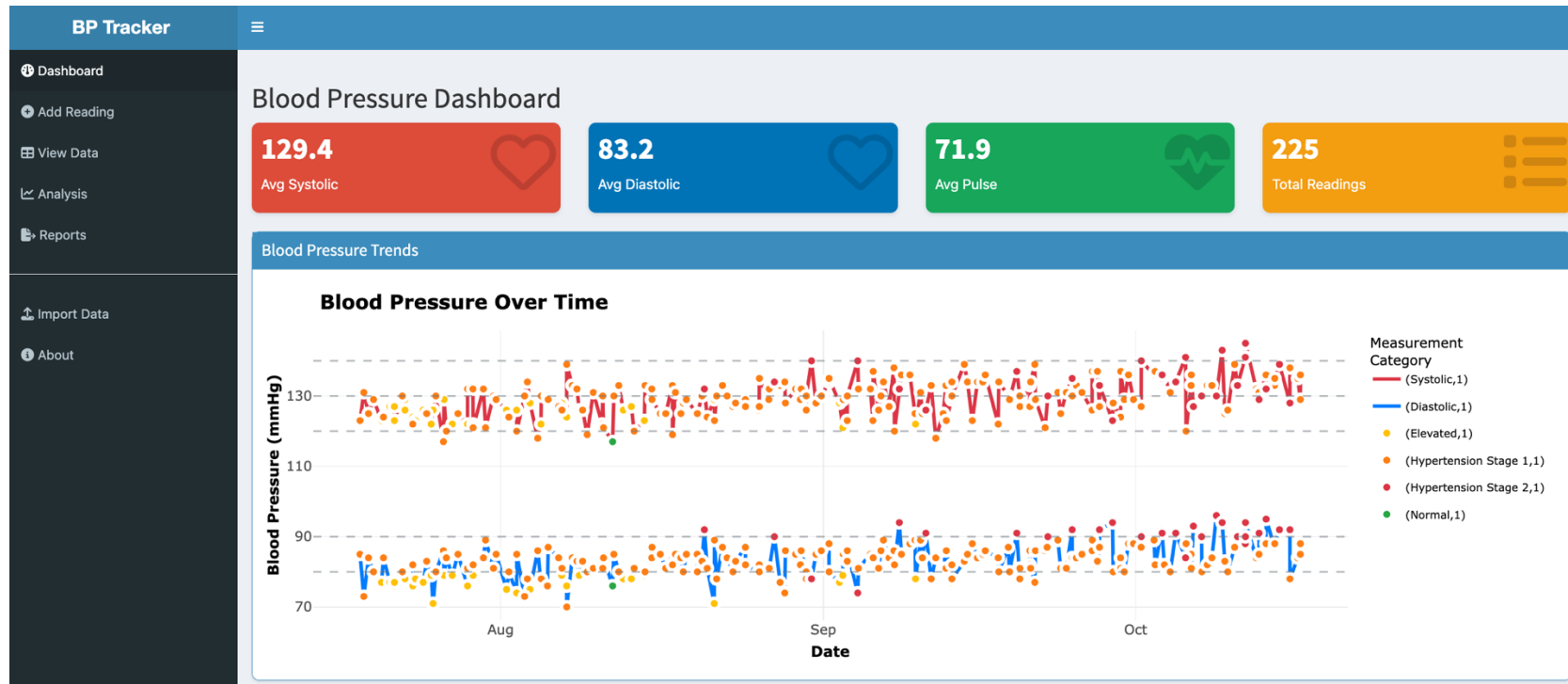
Output:

```
1  # Copilot Instructions
2
3  This file provides guidance for AI coding agents working
   in this R Shiny application codebase.
4
5  ## Project Overview
6
7  **Blood Pressure Analysis Shiny App** – A comprehensive
   application for tracking, analyzing, and visualizing blood
   pressure measurements over time.
```

Prompt:

1. Write a plan for shiny template which helps me analyze my blood pressure
2. Use the plan to create the shiny app

Output:

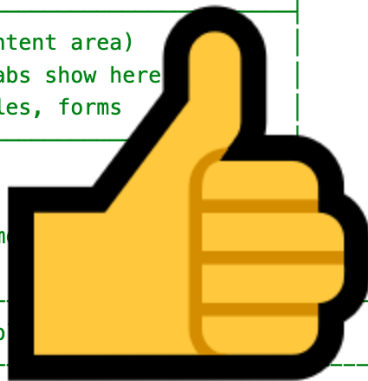


Prompt:

Write me a thorough documentation in app.R, so that I can understand your code

Output:

```
# =====  
# STEP 3: DEFINE THE USER INTERFACE (UI)  
# =====  
# This is what users see and interact with. Think of it as designing a webpage.  
#  
# DASHBOARD STRUCTURE:  
#  
# |-----|  
# | HEADER (top bar with title) |  
# |-----|  
# | SIDEBAR | BODY (main content area) |  
# | (menu) | - Different tabs show here |  
# |         | - Charts, tables, forms |  
# |-----|  
#  
  
ui <- dashboardPage(  
  skin = "blue", # Color theme  
  
  # -----  
  # HEADER: Top bar of the app  
  # -----  
  dashboardHeader(  
    title = "BP Tracker", # Title shown in top-left corner  
    titleWidth = 250      # Width in pixels (matches sidebar width)  
  ),  
  
  # -----  
  # SIDEBAR: Left bar of the app  
  # -----  
  dashboardSidebar(  
    title = "Menu", # Title shown in top-left corner  
    titleWidth = 250 # Width in pixels (matches sidebar width)  
  ),  
  
  # -----  
  # BODY: Main content area  
  # -----  
  dashboardBody(  
    # Content goes here  
  )  
)
```



Is the app ready for production?



- What about software security?
- What about the software architecture?
- What about testing?
- ...

Did this feel like coding?



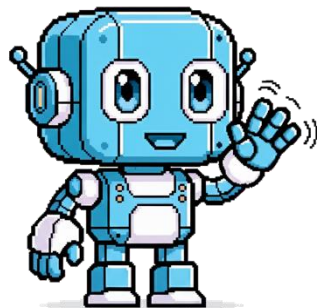
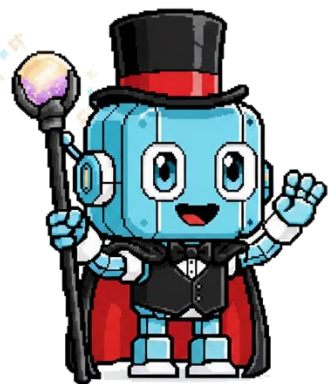
Vibe coder = Project manager

Key takeaways

- You can not trust AI/human developers
 - AI is probabilistic not deterministic
 - AI is as stupid as humans
- Online AI use = Data donation
- Vibe coder = project manager != coder.
 - Vibe coders exist to ensure quality.



Thank you
for your attention!



AI partner in shiny apps

select only the Males that tip over 10%



I'll update the dashboard to show only male customers who tipped more than 10% of their bill.

Enter a message...



Male customers with tips over 10%

```
SELECT *  
FROM tips  
WHERE sex = 'Male'  
AND percent > 0.10
```



Total tippers
136



Average tip
16.9%



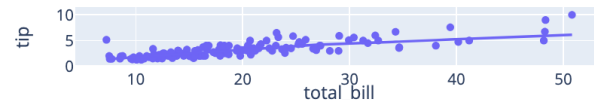
Average bill
\$19.67

Tips data

total_bill	tip	sex	smoker	day	time	size	percent
10.34	1.66	Male	No	Sun	Dinner	3	0.160541586

Viewing rows 1 through 1 of 136

Total bill vs. tip



*TAAFT.com for short

THERE'S AN AI FOR THAT®*

41,134 AI tools for 13,040 tasks and 5,121 jobs

Sponsor:



Rocket

Vibe coding

Search...

⌘ + K



#1 website for AI tools. Used by 60M+ humans.



Trending tools



Generate images



Trending



Characters



Mini tools



New



Rankings



Featured



Collections



Agents



Requests



Tasks



Find A Job



Near me



Forum



Newsletter

<https://theresanaiforthat.com/>

KI in der Softwareentwicklung: Assistenten, Prozesse und Risiken

Sources

+ Add

Discover

Select all sources

AI Pair Programming in 2025: The Good, Bad, and Ugly - Builderio

Einfluss von Künstlicher Intelligenz auf das Embedded Software Engineering

How to Become an Expert in AI-Assisted Coding - A Handbook for Developers

KI-Refactoring: Revolutionäre Code-Optimierung mit Mindverse Studio

KI-generierter Code und Lizenzrisiken: Was Entwickler und Unternehmen beachten...

OpenAI-Codex: Was er ist, wie er funktioniert und wie er verwendet wird - Comet...

Software zur Programmierunterstützung: GitHub Copilot Amazon CodeWhisperer

Welcher KI-Coding-Assistent eignet sich am besten für die Entwicklung komplexer...

Wie man KI bei der Programmierung einsetzt - 12 Best Practices im Jahr 2025

https://www.netzwoche.ch/news/2025-04-04/so-praegt-k-i-die-zukunft-der-soft...

Chat

KI in der Softwareentwicklung: Assistenten, Prozesse und Risiken

10 sources

Die bereitgestellten Quellen beschäftigen sich mit dem Einfluss und der Rolle von Künstlicher Intelligenz (KI), insbesondere generativer KI, im Bereich des Software-Engineerings und des Software-Projektmanagements. Ein Großteil des Materials stammt aus einer Masterarbeit in deutscher Sprache, die systematisch die Auswirkungen von KI auf den gesamten Softwareentwicklungslebenszyklus (SDLC) untersucht, basierend auf Literaturanalysen und Experteninterviews mit Fachleuten von Unternehmen wie Continental und Meta. Ergänzt dazu beleuchten andere Quellen spezifische KI-gestützte Programmierassistenten wie GitHub Copilot und Amazon CodeWhisperer und vergleichen diese mit Open-Source-Alternativen wie TabbyML, wobei Fragen der Datensicherheit und der Anpassung ("Finetuning") im Vordergrund stehen. Die allgemeinen Beobachtungen und Prognosen deuten darauf hin, dass KI routinemäßige Aufgaben automatisieren und die Effizienz steigern wird, während menschliche Expertise für komplexe Aufgaben weiterhin unerlässlich bleibt und neue Rollen entstehen.

Save to note

Video Overview

Audio Overview

Mind Map

Start typing...

10 sources

Welche konkreten Auswirkungen hat Künstliche Intelligenz auf den Softwareentwicklungslebenszyklus und das Projektmanag...

Studio > Report

Briefing-Dokument: Künstliche Intelligenz in Software-Engineering und Projektmanag

Based on 10 sources

Briefing-Dokument: Künstliche Intelligenz in Software-Engineering und Projektmanagement

Executive Summary

Künstliche Intelligenz (KI) hat sich als transformative, jedoch ergänzende Kraft in der Softwareentwicklung etabliert. Sie fungiert als entscheidender Produktivitätsmultiplikator für qualifizierte Entwickler, ersetzt jedoch nicht deren grundlegende Expertise. Die Integration von KI-Tools wie GitHub Copilot, die bereits von 82 % der Entwickler genutzt werden, führt zu signifikanten Effizienzsteigerungen, einschließlich einer bis zu 55 % schnelleren Erledigung von Aufgaben. Der primäre Nutzen liegt in der Automatisierung von Routineaufgaben wie der Erstellung von Boilerplate-Code, der Generierung von Unit-Tests, der Dokumentation und dem Code-Refactoring, was zu einer verbesserten Code-Qualität und kürzeren Entwicklungszyklen führt.

Trotz dieser Vorteile birgt der Einsatz von KI erhebliche Herausforderungen. Dazu gehören technische Limitationen wie kontextuelle Missverständnisse und „Halluzinationen“, bei denen die KI fehlerhaften oder nicht existierenden Code vorschlägt. Ferner bestehen strategische Risiken wie eine übermäßige Abhängigkeit, die zu einem Verlust fundamentaler Programmierfähigkeiten führen kann. Besonders kritisch sind die juristischen und Compliance-Risiken: Da KI-Modelle auf öffentlich zugänglichem Code trainiert werden, besteht die Gefahr unbeabsichtigter Lizenzverstöße (insbesondere durch Copyleft-Lizenzen wie GPL) und Urheberrechtsverletzungen, die weitreichende rechtliche Konsequenzen haben können.

Die Rolle des Entwicklers befindet sich in einem tiefgreifenden Wandel – weg vom reinen Codieren hin zum strategischen Anleiten und Überwachen von KI-Agenten. Kompetenzen im „Prompt Engineering“, im Verstehen von KI-Limitationen und in der übergeordneten Systemarchitektur werden entscheidend. Im Projektmanagement ermöglicht KI die Automatisierung von Reporting, datengestützte Planung auf Basis historischer Analysen und eine effizientere Ressourcennutzung, wobei laut Gartner bis 2030 bis zu 80 % der traditionellen Projektmanagement-Aufgaben automatisiert werden könnten. Der Schlüssel zum Erfolg liegt in der symbiotischen Beziehung zwischen menschlicher Expertise und maschineller Effizienz.

1. Einleitung: Der Paradigmenwechsel im Software-Engineering

Die Softwareentwicklung durchläuft einen Paradigmenwechsel, der durch die rasante Integration von Künstlicher Intelligenz vorangetrieben wird. KI-gestützte Werkzeuge sind nicht länger eine ferne Zukunftsvision, sondern haben sich als alltägliche Hilfsmittel in den Entwicklungsumgebungen (IDEs) etabliert, vergleichbar mit der Syntaxhervorhebung. Eine Studie belegt, dass bereits 82 % der Entwickler täglich oder wöchentlich KI-Tools einsetzen, wobei 59 % sogar drei oder mehr Plattformen parallel nutzen.

Der Gartner Hype Cycle für neue Software-Technologien aus dem Jahr 2023 positioniert das „KI-unterstützte Software-Engineering“ auf dem „Gipfel der überzogenen Erwartungen“ und prognostiziert eine Mainstream-Einführung innerhalb der nächsten zwei bis fünf Jahre. Diese Entwicklung signalisiert einen Wendepunkt: Die zentrale Frage ist nicht mehr, ob KI eingesetzt wird, sondern wie sie strategisch integriert wird, um Qualität, Effizienz und Innovation zu maximieren. Die KI agiert dabei als ein intelligenter Assistent – ein „smarter Praktikant“, der repetitive Aufgaben übernimmt und Entwicklern ermöglicht, sich auf kreative und architektonische Herausforderungen zu konzentrieren.

2. Der Einfluss von KI auf den Softwareentwicklungslebenszyklus (SDLC)

KI-Technologien greifen in jede Phase des traditionellen Softwareentwicklungslebenszyklus ein und bieten Potenziale zur Automatisierung, Optimierung und Qualitätssteigerung.

2.1 Anforderungsanalyse

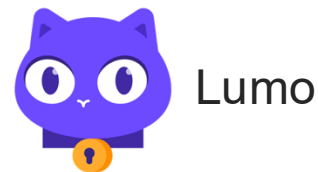
Good report

Bad report

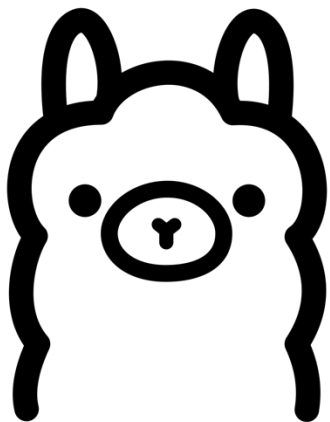
https://notebooklm.google.com

36

What tool can I use? (online)



What tool can I use? (offline)



Ollama



LM studio



AnythingLM



ComfyUI

What tool can I use? (coding/writing)

 **GitHub Copilot**

 **Continue**

 **cline**

**CLAUDE
CODE**

How do I choose my model?



<https://artificialanalysis.ai/?intelligence-tab=coding>



<https://riskrubric.ai/>

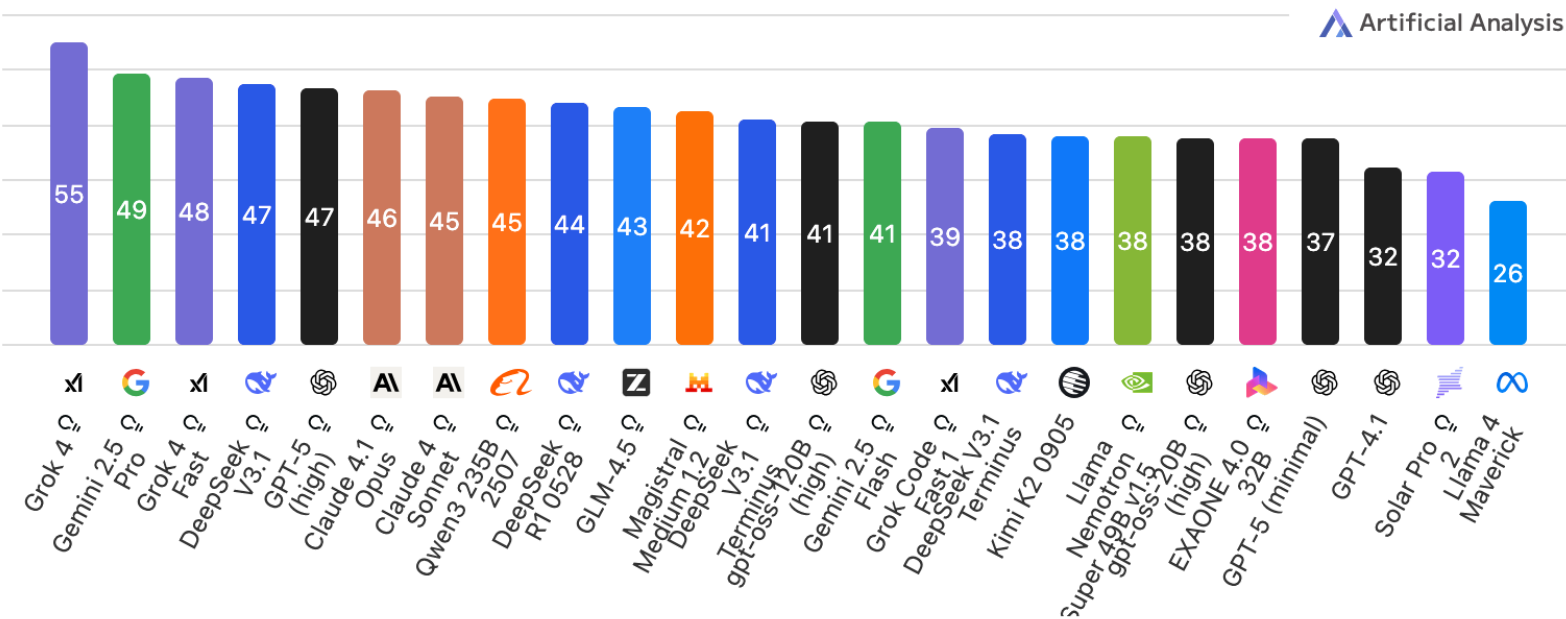


<https://lmarena.ai/leaderboard>

How do I choose my model?

Artificial Analysis Coding Index

Represents the average of coding benchmarks in the Artificial Analysis Intelligence Index (LiveCodeBench, SciCode & Terminal-Bench Hard)



<https://artificialanalysis.ai/?intelligence-tab=coding>