

# From Know-how to How-to

Von LLM Anleitungen zu AI Agenten

**Dr. Florian Steiner**

Claude Code Community Meetup · 11. Februar 2026

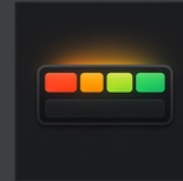
# 4 Produkte. 0 Entwickler.

Vibe Coder & AI Consultant

Produkt Manager der mit AI Produkte baut

CLAUDE.md + CodeRabbit + Session-Handover

Kein Dev-Team — nur Claude Code,  
Infrastruktur und klare Prozesse.



[www.TouchBarFix.com](http://www.TouchBarFix.com)



[www.PatchPilot.dev](http://www.PatchPilot.dev)



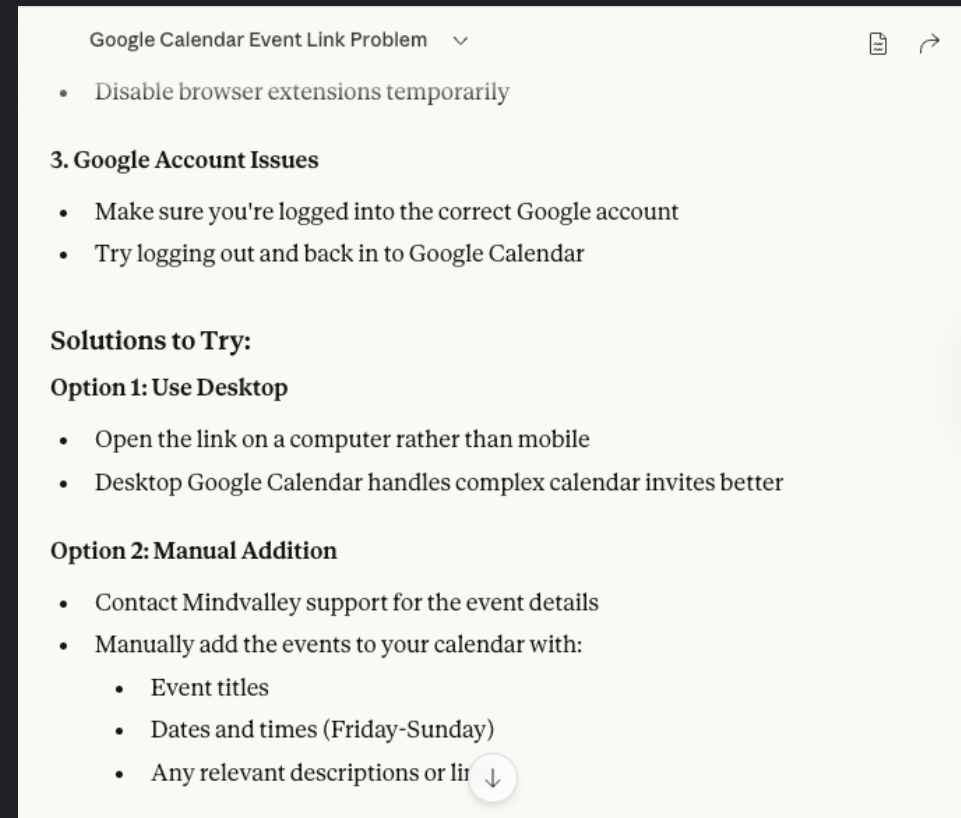
[www.taxfinito.de](http://www.taxfinito.de)

bald verfügbar

[indie-invoice.de](http://indie-invoice.de)

# Das alte Modell: Know-how

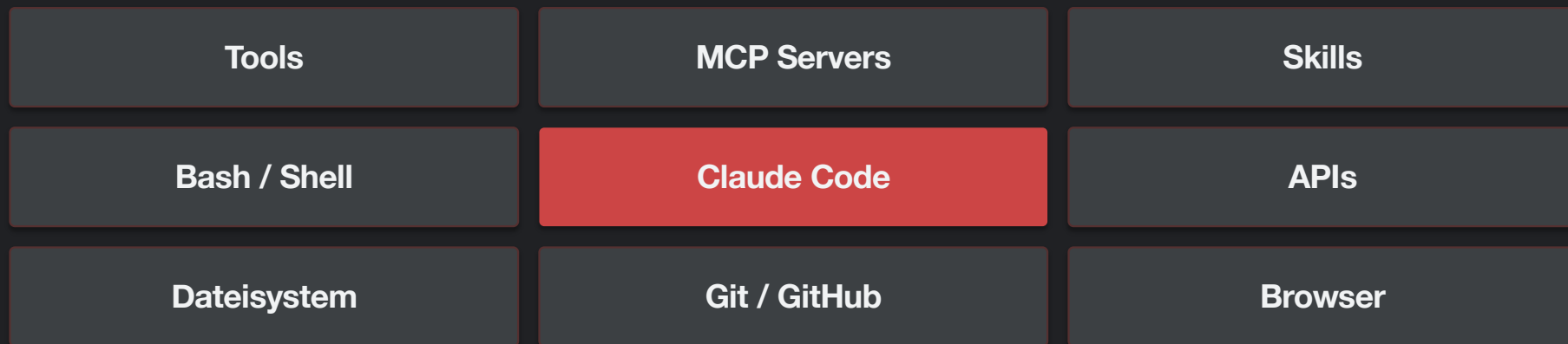
- LLMs liefern Anleitungen, wir führen aus
- Copy & Paste aus dem Chat in andere Programme
- Formatierungsfehler, Screenshots abtippen
- Der Mensch bleibt der Flaschenhals



# Das neue Modell: How-to

- Agent "Implement Invoice Numbering (Issue #6)" completed
  - That's a stale notification from the earlier numbering agent – it already completed and was merged
  - All 4 Phase 2 PRs are merged and verified (198 tests passing on main). The next step is Issue #8 – Ready to proceed when you are.
  - Agent "Implement VIES client (Issue #5)" completed
  - Another stale notification – the VIES agent completed earlier and was merged as PR #11.
  - All Phase 2 work is done and merged. Standing by for Issue #8 (Wire Pipeline) when you're ready.
  - Agent "Implement Tax Engine (Issue #4)" completed
- 
- Claude Code führt Befehle direkt aus
  - Kein Copy & Paste
  - Vom Wissenslieferanten zum autonomen Akteur

# Agent Native Architecture



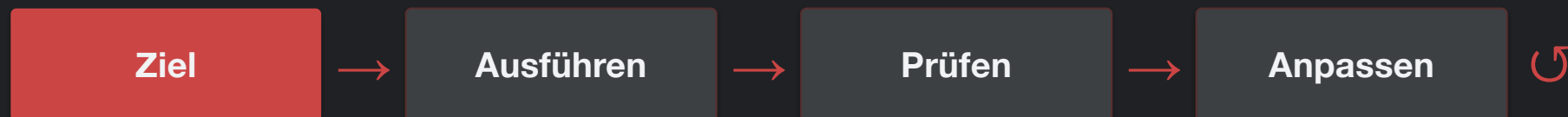
- Agenten können alles ausführen, was der Nutzer ausführen kann
- Kleine, rekombinierbare Bausteine als Basis
- Dynamisch neue Anforderungen mit Prompts erzeugen
- Neue Fähigkeiten mit Skills entwickeln

# Self-Checking: Der Ralph-Wiggum-Loop

Gut beschriebenes Ziel definieren

Claude iteriert selbständig bis zum Ergebnis

Fehler erkennen → korrigieren → weiter



# Beispiel: KI-Avatar-Plattform

**01**

## Anforderung

Kunde braucht interaktiven  
Berater-Avatar für Altersvorsorge

**02**

## Claude Code

Baut Orchestrierungs-Plattform  
autonom (Next.js + HeyGen + GPT-4)

**03**

## Proof of Concept

Funktionsfähiger Demonstrator

# Beispiel: Taxfinito

**01**

## Discovery

30+ Kundeninterviews  
Problem validiert

**02**

## Vibe Coding

SaaS-MVP in 2 Wochen

**03**

## Agent Native

Mac App mit 15 autonomen  
Tools + Skills

# Skills are the new Code

Skills = Textdateien die beschreiben, was Claude tun soll

Kein Programmieren nötig — nur klare Beschreibungen

Claude erkennt und nutzt Skills automatisch

Neue Skills entstehen im Gespräch

„Die meisten Anforderungen brauchen keinen Entwickler mehr.“

# SKILL.md — Beispiel

name: invoice-generator

trigger: "erstelle Rechnung"

steps:

- Kundendaten aus CSV lesen
- Positionen berechnen
- PDF generieren
- Per E-Mail versenden

# Claude Code schließt die Lücke

## Knowing

Fragen beantworten  
Anleitungen liefern  
Im Web recherchieren



## Doing

Dateien erstellen & ändern  
Skills ausführen  
Anwendungen bauen

# Die nächste Stufe: Claude Coworker

The screenshot displays the Claude Coworker interface. On the left, a chat window titled "AI and Double Diamond presentation" shows a conversation where the AI assistant has generated a Python-based solution for a PowerPoint presentation. The chat includes a list of 7 slides and a download link for the presentation. On the right, a presentation viewer displays the first slide, "Der Double Diamond", which outlines the four phases of the Double Diamond process: Discover (Problem erkunden), Define (Problem definieren), Develop (Ideen entwickeln), and Deliver (Lösung liefern). The slide also identifies the Problem Space (orange) and Solution Space (teal), and notes that AI can support every phase. Below the main slide, a detailed view of Phase 1, "Discover: Problem erkunden", is shown, including AI tools and methods like Desk Research and Interview-Transkription, along with a practical example prompt.

AI and Double Diamond presentation

Playwright hat Einschränkungen in dieser Umgebung. Ich wechsele zu einer Python-basierten Lösung mit python-pptx:

Ran 10 commands, edited 2 files, created a file >

Die PowerPoint-Datei wurde erfolgreich erstellt. Da die visuelle Validierung in dieser Umgebung eingeschränkt ist, verifiziere ich die Dateistruktur:

Ran 2 commands, updated todo list >

Die Präsentation ist fertig. Hier ist deine PowerPoint-Datei:

[Präsentation herunterladen](#)

**Inhalt der 7 Folien:**

1. **Titelfolie** - AI im Double Diamond mit Zielgruppen-Badge
2. **Übersicht** - Double Diamond Diagramm mit Problem Space (orange) und Solution Space (teal)
3. **Discover** - Research-Tools wie ChatGPT, Otter.ai und praktische Prompt-Beispiele
4. **Define** - Persona-Entwürfe, Journey Maps und Problem Statements
5. **Develop** - Ideation, Rapid Prototyping mit Midjourney/v0
6. **Deliver** - Usability-Analyse, Design-to-Code, Dokumentation
7. **Key Takeaways** - Drei Prinzipien: AI als Beschleuniger, Mensch bleibt zentral, Immer validieren

AI im double diamond Presentation · PPTX

Open in Microsoft PowerPoint

Page 1 / 7

## Der Double Diamond

Vier Phasen, zwei Denkweisen: Divergent und Konvergent

AI kann in JEDER Phase unterstützen

**PROBLEM SPACE**

**SOLUTION SPACE**

**DIVERGENT**  
Optionen erweitern, Möglichkeiten erkunden

**KONVERGENT**  
Fokussieren, priorisieren, entscheiden

Page 2 / 7

**PHASE 1 Discover: Problem erkunden**

Divergent denken - Breite Recherche und Nutzerverständnis aufbauen

**AI-TOOLS UND METHODEN**

**Desk Research beschleunigen**  
ChatGPT, Perplexity fuer schnelle Markt- und Wettbewerbsanalysen

**Interview-Transkription**  
Otter.ai, Whisper fuer automatische Verschriftlichung

**PRAXIS-BEISPIEL**

PROMPT  
"Analysiere diese 15 Nutzerinterviews und identifiziere die Top-5 Pain Points mit Zitaten als Belege."

**Wichtig dabei:**

# Mitmachen

- Vibe Coding Scaffold Template
- CLAUDE.md + /start + /wrap + CodeRabbit + Session-Handover

The screenshot displays the GitHub repository interface for 'vibe-coding-scaffold'. At the top, it shows the repository name, 'Public' status, and options to Pin or Watch. Below this, there are navigation elements for the 'main' branch, showing 1 branch and 0 tags, along with a search bar and buttons for 'Add file' and 'Code'. The commit history table lists several files and their commit details:

File	Commit Message	Author	Time
sessions	Initial commit: Vibe Coding Scaffold Template	ProduktEntdecker and claude	2 days ago
.coderrabbit.yaml	Initial commit: Vibe Coding Scaffold Template	ProduktEntdecker and claude	2 days ago
CLAUDE.md	Initial commit: Vibe Coding Scaffold Template	ProduktEntdecker and claude	2 days ago
LEARNINGS.md	Initial commit: Vibe Coding Scaffold Template	ProduktEntdecker and claude	2 days ago
ONE-THING.md	Initial commit: Vibe Coding Scaffold Template	ProduktEntdecker and claude	2 days ago
README.md	fix: Set correct repo URL in Quick Start	ProduktEntdecker and claude	2 days ago
TOOLS.md	Initial commit: Vibe Coding Scaffold Template	ProduktEntdecker and claude	2 days ago

Below the table, the README section is visible, titled 'Vibe Coding Scaffold'. The text in the README reads: 'Ein Template für produktionsreifes Vibe Coding mit Claude Code.'

<https://github.com/ProduktEntdecker/vibe-coding-scaffold>

# Vernetzen

[linkedin.com/in/floriansteiner](https://www.linkedin.com/in/floriansteiner)



[www.produktentdecker.com](https://www.produktentdecker.com)