

Bionic Logo

AI Coding aka Vibe Coding

Eine Einführung in die Zukunft der Softwareentwicklung

Bedo Hamza

HSLU | 02. Februar 2026

Agenda

- Was ist AI Coding / Vibe Coding?
- Die Evolution der Programmierung
- Aktuelle AI Coding Tools im Überblick
- Live-Demos und Praxisbeispiele
- Chancen und Risiken
- Die Zukunft der Softwareentwicklung
- Q&A Session

Was ist "Vibe Coding"?

"Vibe Coding ist ein Programmier-Ansatz, bei dem man sich voll auf die Intuition einlässt, die Welle der KI-generierten Vorschläge reitet und vergisst, dass der Code überhaupt existiert."

— Andrej Karpathy, Februar 2025

Der neue Ansatz:

- Natürliche Sprache statt Syntax
- Intention beschreiben, nicht implementieren
- AI als Co-Pilot, nicht als Werkzeug

Das Ergebnis:

- Schnellere Prototypen
- Niedrigere Einstiegshürden
- Fokus auf Kreativität

Die Evolution der Programmierung

1950er

Maschinencode & Assembly

Direkte Hardware-Programmierung

1970er-90er

Hochsprachen (C, Java, Python)

Abstraktion von der Hardware

2000er-2010er

Frameworks & Libraries

Wiederverwendbare Komponenten

2020er

AI-Assisted Coding

Copilot, ChatGPT, Claude

2025+

Vibe Coding / AI Agents

Natürliche Sprache → Code

AI Coding in Zahlen

92%

der Entwickler nutzen
bereits AI-Tools

► [Klicken für Details](#)

55%

schnellere Task-
Completion

► [Klicken für Details](#)

46%

des Codes wird AI-
generiert

► [Klicken für Details](#)

1.3M+

Copilot Business
Nutzer

► [Klicken für Details](#)

Quellen: GitHub Survey 2024, Stack Overflow Developer Survey 2024

⚠ Kritische Würdigung: Diese Zahlen stammen primär aus GitHubs eigenen Untersuchungen zur Effektivität von Copilot (Marketing-Daten).

AI Coding Tools - Übersicht



GitHub Copilot

IDE-Integration, Code-Completion



Cursor

AI-First Code Editor



Claude Code

Agentic Coding Terminal



Windsurf

AI-powered IDE von Codeium



Lovable.dev

Full-Stack App Builder



v0.dev & Bolt

UI/Component Generation

Kategorien von AI Coding Tools

Code Assistenten

- GitHub Copilot
- Amazon CodeWhisperer
- Tabnine

AI-First IDEs

- Cursor
- Windsurf
- Zed (mit AI)

Agentic Coding

- Claude Code
- Devin
- OpenHands

App Builder

- Lovable.dev
- v0.dev
- Bolt.new

Tools nach Skill-Level



02

Die Tools im Detail

GitHub Copilot • Cursor • Claude Code • Windsurf • Lovable.dev

GitHub Copilot

Your AI pair programmer

✓ Vorteile

Beste IDE-Integration

Schnelle Inline-Suggestions

GitHub-Ökosystem

Multi-Language Support

Enterprise-ready

✗ Nachteile

Kostenpflichtig (\$19/Monat)

Begrenzte Kontextgrösse

Kein Multi-File Editing

Datenschutz-Bedenken

GitHub Copilot

Cursor

The AI-first Code Editor

✓ Vorteile

Multi-File Editing

Codebase-Kontext

VS Code-basiert

Chat + Composer

Schnelle Iteration

✗ Nachteile

Pro: \$20/Monat

Lernkurve für Features

Manchmal instabil

Token-Limits

Cursor

Claude Code



Agentic Coding im Terminal

✓ Vorteile

✓ Agentic Workflow

✓ Grosser Kontext (200k)

✓ Multi-File nativ

✓ Git-Integration

✓ Anthropic-Qualität

✗ Nachteile

✗ Terminal-only

✗ API-Kosten

✗ Keine GUI

✗ Lernkurve

Claude Code

Windsurf

AI IDE von Codeium

✓ Vorteile

Cascade (Agentic)

Flows Feature

Schnelle Completion

Free Tier grosszügig

Multi-Model Support

✗ Nachteile

Relativ neu

Kleinere Community

Weniger Extensions

Occasional Bugs

Windsurf

Lovable.dev



Full-Stack App Builder

✓ Vorteile

Keine Coding-Kenntnisse
nötig

Full-Stack aus Prompt

Supabase-Integration

Instant Deployment

Visual Editing

✗ Nachteile

Limitierte Kontrolle

Vendor Lock-in

Kosten bei Skalierung

Weniger Flexibilität

Lovable.dev

Tool-Vergleich

Tool	Typ	Preis	Bestes für
GitHub Copilot	Assistant	\$19/mo	IDE-Integration
Cursor	AI IDE	\$20/mo	Multi-File Editing
Claude Code	Agentic	API-based	Komplexe Tasks
Windsurf	AI IDE	Free/Pro	All-in-One
Lovable.dev	Builder	Free/Pro	No-Code Apps

Unser Vibe Coding Workflow

□ Setup Phase

□ **Idee entwickeln**
Claude Chat



□ **Produkt definieren**
Walter PO MCP Server (dinbrudi.com) + Claude Code



□ **Design erstellen**
Figma Make

□ Iterativer Zyklus

REPEAT

⚡ **Implementieren**
Claude Code (Terminal)



□ **Testen**
Regelmässig prüfen!



□ **Commit & Version**
GitHub (Rollback möglich!)

🔄 Wiederholen bis Feature fertig



□ **Deployen**
Xcode / App Store / Web

⚠ Wichtig: GitHub verstehen für Versionierung und Fehler-Recovery

Unsere Vision: Bionic Product Development

Wir möchten diesen Workflow so **vereinfachen**, dass er von einer **grossen Anzahl an Kunden** genutzt werden kann.

Bionic Logo

bionic-product-development.com

Die Plattform für AI-gestützte Produktentwicklung

[Als Beta-Tester anmelden](#)

Sei dabei und gestalte die Zukunft der Produktentwicklung mit!

03

Live Demo

SRF News App mit Vibe Coding

Demo: SRF News App

Beispiel-Prompts für jeden Workflow-Schritt · API: developer.srgssr.ch (SRGSSR Articles API)

□ 1. Idee entwickeln Claude Chat

"Ich möchte eine einfache iOS News App bauen, die SRF News anzeigt. Die SRG SSR bietet eine öffentliche Articles API an. Welche Features wären sinnvoll für ein MVP?"

□ 2. Produkt definieren Walter PO MCP + Claude Code

"Erstelle ein neues Produkt 'SRF News App' mit Features: News Feed Ansicht, Artikel Detail, Kategorien Filter. Erfasse die Tasks für das erste Feature."

□ 3. Design erstellen Figma Make

"Erstelle ein minimalistisches Design für eine News App mit: Header mit SRF Logo, Card-basierter News Feed, Pull-to-refresh, Dark Mode Support."

⚡ 4. Implementieren Claude Code (Terminal)

"Erstelle eine SwiftUI iOS App mit MVVM Architektur. Implementiere einen API Service für die SRGSSR Articles API (Base URL: api.srgssr.ch). Zeige die News in einer scrollbaren Liste mit AsyncImage für Thumbnails."

□ 5. Testen & Deploy Xcode + GitHub

"Teste die App im Simulator, committe nach jedem funktionierenden Feature. Bei Fehlern: git diff prüfen, ggf. git checkout zur letzten Version."

Chancen von AI Coding



Produktivität

10x schnellere Entwicklung für
Prototypen und MVPs



Lernen

AI erklärt Code und Best Practices
in Echtzeit



Demokratisierung

Mehr Menschen können Software
erstellen



Iteration

Schnelles Experimentieren und
Testen neuer Ideen



Debugging

AI findet und erklärt Bugs
automatisch



Dokumentation

Automatische Code-
Dokumentation

Risiken & Herausforderungen

⚠ Technische Risiken

- Halluzinationen & Fehler im Code
- Sicherheitslücken werden übersehen
- Veralterte Patterns/Libraries
- Schwer zu debuggen

⚠ Professionelle Risiken

- Skill-Atrophie bei Entwicklern
- Über-Vertrauen in AI
- Copyright & Lizenzfragen
- Job-Markt Veränderungen

📦 Fragmentierte Tool-Landschaft

- Viele Tools, wenig Standardisierung
- Steile Lernkurve für effektives Arbeiten

⚡ Rasante Entwicklung

- Tools ändern sich monatlich
- Wissen veraltet schnell

- Meine Skills: über 1 Jahr Aufbauarbeit

- Kontinuierliches Lernen erforderlich

Best Practices für AI Coding



Code Review

Immer AI-generierten Code prüfen und verstehen



Testing

Tests schreiben (lassen) und ausführen



Lernen

AI als Lernwerkzeug nutzen, nicht nur als Generator



Spezifische Prompts

Je klarer der Prompt, desto besser das Ergebnis

Die Zukunft des Codings

Kurzfristig (2025-2026)

- Bessere Multi-File Agents
- Mehr IDE-Integrationen
- Spezialisierte Coding-Modelle

Mittelfristig (2026-2028)

- Autonome Bug-Fixing Agents
- AI-gesteuerte Code Reviews
- Natural Language Interfaces

Langfristig (2028+)

- Software aus Spezifikation
- Self-healing Systems
- AI-AI Kollaboration

"The best code is the code you don't have to write."

Meine Empfehlungen

Für Anfänger:

- **Lovable.dev** - Schnelle Ergebnisse
- **Cursor** - Guter Editor zum Lernen
- Grundlagen parallel lernen!

Für Profis:

- **Claude Code** - Komplexe Projekte
- **Cursor/Windsurf** - Daily Driver
- Copilot als Ergänzung

Tipp: Starte mit einem Tool, meistere es, dann erweitere dein Arsenal.

?

Fragen & Diskussion

Was möchtet ihr wissen?

📱 Live Produkte - In Production & von Kunden genutzt

📱 BRONZE WINNER

DinBrudi Icon
DinBrudi

Ersti ToDo App uf Schwiizerdütsch

Best of Swiss Apps - Kategorie Innovation

iOS

Android

[App Store →](#)

Bru Icon
Bru your todo

Internationale Voice First ToDo App

Sprachgesteuerte Produktivität

iOS

[App Store →](#)

DinBrudiTeam Icon
DinBrudiTeam

ToDo und Protokoll Plattform für Teams

Team Collaboration & Workflows

Web App

[dinbrudi.com →](https://dinbrudi.com)

📱 BRONZE WINNER

Bionic VYW Icon
bionic - visualize your wins

Apple Vision Pro Success Tracking App

Best of Swiss Apps - Extended Interactions

visionOS

[getbionic.app →](https://getbionic.app)

□ **Von der Idee zum Award-Winner:** Alle 4 Apps wurden mit AI-gestützter Entwicklung und dem Bionic-Prozess umgesetzt - von Konzept bis Launch in Wochen statt Monaten.

Bionic Logo

Vielen Dank!

Bedo Hamza

✉ bedo@hmz.swiss

HSLU | 02. Februar 2026