

Future Lab: AI Coding

Mit Coding-Agenten eigene Webanwendungen planen, entwickeln und systematisch prüfen

Bring deinen B2B-Use Case vom ersten Entwurf zum testbaren Webprototyp. Lerne, Anforderungen präzise zu formulieren, Kontext für Coding-Agenten aufzubauen und Ergebnisse mit Tests und Reviews abzusichern. Am Ende nimmst du einen dokumentierten Prototyp samt Kontextpaket, Testnachweisen, Lernlogbuch und Entwicklungsplan mit.

Im Januar geht es los – lass dich benachrichtigen

Das erste Future Lab wird im Januar 2027 starten. Weitere Durchgänge folgen im März und im Juni. Die konkreten Termine veröffentlichen wir in Kürze. Lass dich einfach benachrichtigen, sobald das Future Lab buchbar ist.

Modul 1: Webinar

4 h

Lab-Session 1: Vibe Coding erleben

- Erleben, wie mit einem KI-Coding-Werkzeug eine kleine Webanwendung entsteht.
- Starte mit dem Aufbau deines eigenen Lernlogbuches für deinen Use Case.

Modul 2: Selbstlernphase

6 - 8 h

Hands-On-Phase 1: Vibe Coding selbst erleben & den eigenen Use Case identifizieren

- Dein Lernlogbuch eigenständig weiterentwickeln
- Vergleiche, wie unterschiedlich Agenten auf einen allgemeinen Wunsch und einen präzisen Arbeitsauftrag reagieren

Modul 3: Webinar

4 h

Lab-Session 2: Deinen Use Case schärfen & validieren

Du wählst methodisch einen überschaubaren B2B-Anwendungsfall und schärfst deine Idee, u. a.

- durch die Wahl und Beschreibung deiner Nutzergruppe
- Definition von Auslösern und Ergebnissen
- Untersuchung von drei vergleichbare Lösungen
- Validierung deiner Idee im Peer-Austausch

Modul 4:
Selbstlernphase

6 - 8 h

Hands-On-Phase 2: Den Kontext für deinen Use Case aufsetzen

Deine in der 2. Lab-Session geschärfte & validierte Use Case-Idee arbeitest du eigenständig weiter aus.

Dazu gehören:

- dein One-Pager
- das Moodboard (designs)
- ggf. Wireframes oder Mocks
- die Architekturskizze/md.

Ein strukturiertes Interview mit einem Sprachmodell hilft dir, offene Annahmen sichtbar zu machen.

Die Hands-On-Phase schließt damit ab, dass du

- deinen Produktkontext konkretisierst
- deinen bisherigen Prototyp exportierst
- den technischen Stand anhand von "Dokument 0" prüfst.

Das vollständige technische Kontextpaket (Agents.md, architecture plan.md etc.) entsteht in der nächsten, 3. Hands-On-Phase.

Modul 5:
Webinar

4 h

Lab-Session 3: Kontext Engineering - von der Produktidee zum umsetzbaren Plan

Du lernst das Vorgehen für dein Projekt kennen:

- Überführen in die Arbeitsumgebung, in der Coding-Agenten mit deinem Repository arbeiten
- Den Projektstand mit Git sichern und strukturiert weiterentwickeln
- Arbeitspakete in einer PLAN.md ordnen, einzelne Features mit klaren Grenzen und Akzeptanzkriterien beschreiben
- AGENTS.md oder CLAUDE.md: gezielt die Arbeitsregeln für deinen Coding-Agenten festhalten
- Skills, Hooks und MCP einsetzen, wo sie eine konkrete Aufgabe vereinfachen

Darüber hinaus steigst du in die Grundlagen für kontrollierte Loops und Tests ein.

Modul 6:
Selbstlernphase

6 - 8 h

Hands-On-Phase 3: Das Kontextpaket für deinen Use Case erarbeiten & Vorbereitung Challenge I

Du erstellst die Arbeitspakete (Kontextpaket) für deinen eigenen Use Case:

- PLAN.md
- Einzelne Features mit klaren Grenzen und Akzeptanzkriterien
- AGENTS.md oder CLAUDE.md: Arbeitsregeln für deinen Agent
- Skills, Hooks und MCP, wo sie eine konkrete Aufgabe vereinfachen

Dabei wendest du in die Grundlagen für kontrollierte Loops und Tests an.

Modul 7: Selbstlernphase

6 - 8 h

Challenge I: Kontext Engineering für deinen Use Case & Iteration

In der 1. Challenge reichst du dein Kontextpaket (aus der Hands-On-Phase 3) ein und erhältst schriftliches Feedback zu Problemverständnis, Umfang, Annahmen und Prüfbarkeit.

Erst nach diesem Feedback beginnt die Implementierung mit neu generiertem Repository-Code.

Vor der nächsten Lab-Session:

Loop und Graph Engineering: Coding-Agenten kontrolliert arbeiten lassen

- Du baust in einem ca. dreitägigen Ausführungsfenster die erste funktionsfähige Version deines Projekts, deinen MVP.
- Dabei wendest du in die Grundlagen für kontrollierte Loops und Tests an.

Modul 8: Webinar

4 h

Lab-Session 4: Loop und Graph Engineering - Coding-Agenten kontrolliert arbeiten lassen

- Iteration deines MVPs: Anforderungen spezifizieren, Aufgaben delegieren, Ergebnisse prüfen und daraus lernen.
- Akzeptanzkriterien mit passenden Tests verknüpfen
- Test-Driven Development als Grundlage für überprüfbare Delegation.
- Builds, Linting und Tests: Änderungen und Verhalten kontrollieren, bevor du sie akzeptierst.
- Erkennen, wann du einen Loop stoppst und zunächst Anforderungen, Kontext oder Agentenregeln korrigieren solltest.
- Peer-Review: Kernablauf zeigen, die zugehörigen Anforderungen und einen fehlgeschlagenen Loop beschreiben, Fallstricke identifizieren und brauchbare Arbeitsmuster für die nächste Projektphase definieren.

**Modul 9:
Selbstlernphase**

6 - 8 h

Hands-On-Session 5: Deinen Prototyp weiterentwickeln & den Betrieb vorbereiten

- MVP entlang kleiner Anforderungen weiterentwickeln und Änderungen systematisch überprüfen.
- Blick auf Hosting, Deployment und der Trennung von Entwicklungs-, Test- und Produktionsumgebung.
- HTTPS, Authentifizierung, Rollen und eine sichere Verwaltung von Zugangsdaten umsetzen
- Bedeutung von Backups, Protokollierung, CI /CD und Monitoring für das eigene Projekt prüfen.
- Datenverarbeitung deiner Anwendung beschreiben, ein Löschkonzept entwickeln und dessen Umsetzung mit synthetischen oder freigegebenen, nicht sensiblen Daten testen.
- Kostenkontrolle: Tokenlimits setzen, Kosten im Blick behalten und prüfen, welche Inhalte an Modellanbieter übermittelt werden.
- Offene Maßnahmen im priorisierten **Production-Readiness-Plan** festhalten.

&

Ergebnisse belegen, präsentieren und nächste Schritte entscheiden

Erarbeitung deines Product Evidence Package

- es enthält deinen lauffähigen Prototyp
- das Repository oder einen vollständigen Export
- eine verständliche Startanleitung
- umgesetzte Anforderungen
- Testnachweise
- einen Security- und Kostencheck

Dieses Paket gibst du als Challenge II ab.

**Modul 10:
Selbstlernphase**

6 - 8 h

Challenge II: Product Evidence Package für deinen Use Case & Vorbereitung deines PitchesIn der **2. Challenge** stellst du dein **Product Evidence Package** zusammen und gibst es ab.

- Es enthält deinen lauffähigen Prototyp
- das Repository oder einen vollständigen Export
- eine verständliche Startanleitung
- umgesetzte Anforderungen
- Testnachweise
- Security- und Kostencheck

Nach dem schriftlichen Trainer:innenfeedback und vor der abschließenden Lab-Session:

- Gezieltes Überarbeiten der Punkte, die den größten Nutzen bringen.
- Vorbereitung auf deinen strukturierten Use Case-Pitch in der letzten Lab-Session:
 - Pitch und reproduzierbarer Demo
 - Wichtige Entscheidungen erläutern
 - Einen korrigierten Loop und bekannte Grenzen erklären
 - Benennen sinnvoller nächster Schritt für dein Projekt – beispielsweise einen Nutzertest, eine Integration oder technische Härtung

Modul 11: Webinar

4 h

Lab-Session 5: Use Case-Pitch & Investing Game

Use Case-Pitch:

- Strukturierte Projektvorstellung
- Reproduzierbare Demo
- Erläutern wichtiger Entscheidungen, eines korrigierten Loops, bekannter Grenzen
- Benennen sinnvoller nächster Schritt für dein Projekt – beispielsweise einen Nutzertest, eine Integration oder technische Härtung.

Investing Game:

Im Investing Game verteilt ihr fiktive Budgets auf die Projekte der anderen. Im Mittelpunkt stehen nachvollziehbarer Nutzen, belastbare Nachweise und eine plausible Weiterentwicklung.

Inhalt & Ablauf des Future Labs

1. Vibe Coding erleben und deinen Anwendungsfall schärfen

- Auswahl eines eigenen, bearbeitbaren B2B-Anwendungsfalls mit Beschreibung von Nutzergruppe, Auslöser und Ergebnis.
- Research: Analyse von drei vergleichbaren Lösungen und Peer-Austausch zur Reflexion der Idee.
- Artefakte erstellen: One-Pager, Moodboard, erste Wireframes, Architekturskizze. Strukturiertes Interview mit einem Sprachmodell zur Klärung offener Annahmen.

2. Kontext Engineering: Von der Produktidee zum umsetzbaren Plan

- Überführen deines Projektes in die Arbeitsumgebung, in der Coding-Agenten mit dem Repository arbeiten. Projektstand mit Git sichern.
- PLAN.md: Arbeitspakete strukturieren, Features mit klaren Grenzen und Akzeptanzkriterien beschreiben.
- AGENTS.md/CLAUDE.md: Regeln und Leitplanken für den Agenten festhalten, gezielter Einsatz von Skills, Hooks und MCP.
- **Challenge I:** Kontextpaket einreichen & individuelles Feedback zu Problemverständnis, Umfang, Annahmen und Prüfbarkeit erhalten.
- Implementierung startet erst nach dem Feedback mit neu generiertem Repository-Code.

3. Loop und Graph Engineering: Coding-Agenten kontrolliert arbeiten lassen

- Bau eines funktionsfähigen MVP in kurzen, kontrollierten Schleifen (Anforderungen präzisieren, Aufgaben delegieren, Ergebnisse prüfen und daraus Lernen).

- Verknüpfung von Akzeptanzkriterien mit passenden Tests: Test-Driven Development als Basis für überprüfbare Delegation.
- Automatisiertes Feedback durch Builds, Linting und Tests. Änderungen und Verhalten vor dem Merge kontrollieren.
- Peer-Review: Kernablauf, Anforderungen und ein fehlgeschlagener Loop zeigen, Fallstricke und Muster identifizieren.

4. Deinen Prototyp weiterentwickeln und den Betrieb vorbereiten

- Iterative Weiterentwicklung des MVP entlang kleiner Anforderungen, systematische Überprüfung von Änderungen.
- Betrieb: Hosting, Deployment und Trennung von Entwicklungs-, Test- und Produktionsumgebung vorbereiten.
- Sicherheit: HTTPS, Authentifizierung, Rollen, sichere Geheimnisverwaltung umsetzen.
- Betriebsfähigkeit: Backups, Protokollierung, CI/CD, Monitoring auf Relevanz prüfen, Datenverarbeitung dokumentieren und Löschkonzept entwickeln und testen.
- Ressourcenkontrolle: Tokenlimits setzen, Kosten im Blick behalten, übermittelte Inhalte an Modellanbieter prüfen, offene Maßnahmen im Production-Readiness-Plan priorisieren.

5. Ergebnisse belegen, präsentieren und nächste Schritte entscheiden

- **Challenge II:** Product Evidence Package zusammenstellen (lauffähiger Prototyp, Repository/Export, Startanleitung, umgesetzte Anforderungen, Testnachweise, Security- und Kostencheck).
- Nach individuellem Trainer:innenfeedback: gezielte Überarbeitung der wichtigsten Punkte.
- **Vorbereitung für die Abschluss-Session:** Pitch des eigenen Use Case & reproduzierbare Demo, wichtige Entscheidungen, korrigierter Loop und bekannte Grenzen erläutern. Nächsten sinnvollen Schritt benennen (z. B. Nutzertest, Integration, technische Härtung).
- **Investing Game:** Fiktive Budgets für die Use Cases der Peers nach Nutzen, Nachweisen und plausibler Weiterentwicklung vergeben.

Lernumgebung

In deiner Online-Lernumgebung findest du nach deiner Anmeldung nützliche Informationen, Downloads und Extra-Services zu dieser Qualifizierungsmaßnahme.

Dein Nutzen

- **Geschäftsideen in klare Anforderungen übersetzen:** Du grenzt deinen Use Case ein, schärfst Zielgruppe und Nutzen und beschreibst, was deine Anwendung leisten soll – und was bewusst nicht.
- **Kontext gestalten statt nur Wünsche formulieren:** Du bereitest Informationen, Projektregeln und Akzeptanzkriterien so auf, dass Coding-Agenten konkrete, überprüfbare Aufgaben bearbeiten können.
- **Coding-Agenten mit Loop und Graph Engineering gezielt steuern:** Du lernst, Entwicklungsaufgaben in kleine Schritte zu zerlegen, an Agenten zu delegieren und anhand von Tests und Reviews über die nächsten Schritte zu entscheiden. Du behältst die Verantwortung für den Prozess und das Ergebnis.
- **Qualität und Grenzen nachvollziehbar beurteilen:** Du prüfst nicht nur, ob deine Anwendung etwas anzeigt, sondern auch ob Verhalten, Anforderungen und Nachweise zusammenpassen.
- **Die Weiterentwicklung fundiert planen:** Du berücksichtigst Datenschutz, Sicherheit und Tokenkosten und hältst fest, welche Schritte deinem Prototyp noch zur Produktionsreife fehlen.

Methoden

Das bringst du mit:

- Erste Erfahrung mit KI Coding-Werkzeugen wie **Claude Code** oder **Codex** oder mit Prompt-to-App-Werkzeugen wie **Lovable** oder **Bolt**
 - Häufig fehlt dir noch der Arbeitsprozess, der aus einem schnellen Entwurf eine planbare und testbare Anwendung macht.
- Technisches Interesse, die Bereitschaft zur **Arbeit im Terminal** und einen eigenen, ausreichend kleinen Use Case

- Lust **eigenverantwortlich** an deinem Use Case zu arbeiten
- **Bereitschaft** und **Zeit**, deinen eigenen Use Case in den Hand On-Phasen auszuarbeiten und zu entwickeln.
- Tools & Lizenzen (s. u.), um deinen Use Case technisch umzusetzen.
- Synthetischen oder freigegebenen, nicht sensiblen **Daten** für deinen Use Case
- Die **Verantwortung** für Use Case, Daten, Repository, und Tool- & Lizenz-Kosten.

Out of Scope:

- Use Cases für Desktop-Anwendungen und Vorhaben mit sensiblen Produktionsdaten
- Zu große Use Cases, die im Zeitraum des Future Labs nicht umgesetzt werden können
- Alles, was eine an die Pilotierung anschließende unternehmensspezifische Weiterarbeit erfordert, wie bspw. Hosting und Deployment

Disclaimer

- Der fertige und gefeedbackte Use Case aus dem Future Labs ist keine Produktionsfreigabe.

Erforderliche Tools und Zugänge:

- Die Werkzeugwahl ist offen
- **Deine Arbeitsumgebung muss zum Start funktionieren**; die erste Lab-Session ist kein Installationsworkshop
- **Empfohlenes technisches Setup:**
 - Lovable Pro oder Bolt Pro, aktuell jeweils ab 25 US-Dollar/monatlich.
 - Für die weitere Arbeit im Repository:
 - Mindestens Claude Pro inklusive Claude Code oder ChatGPT Plus inklusive Codex, aktuell jeweils ca. 20 \$ /monatlich
 - Empfohlen: Claude Max inklusive Claude Code oder ChatGPT Pro inklusive Codex, aktuell jeweils ca. 100 \$ /monatlich
 - Der jeweilige Chat-Zugang ist damit bereits abgedeckt.
 - Cloud-Accounts und Adminrechte:
 - Entscheidend ist eine getestete Arbeitsumgebung mit Repository-Zugriff. Ein Account allein reicht dafür noch nicht.
 - Für den lokalen Weg müssen Git, Coding-Werkzeug (IDE) und die benötigte Laufzeit eingerichtet sein, oder Cloud IDE.
 - Pauschale Adminrechte sind nicht zwingend erforderlich. Die Installation und Nutzung müssen durch die Teilnehmenden oder ihre IT ermöglicht werden. Ein vorbereiteter Cloud-Arbeitsplatz kann ebenfalls passen.
 - Zur Vorbereitung steht dir ein optionaler Kurs 0 mit Checkliste und Lernressourcen zu Git, GitHub, Supabase, Vercel, lokalem Setup und Coding-Arbeitsumgebungen zur Verfügung. Mit passenden Vorkenntnissen kannst du diese Vorbereitung überspringen.
 - Budget für 1 – 2 Abrechnungsmonate: ca. 100 – 150 €, einschließlich einer Reserve für zusätzliche Nutzung.
 - Das ist ein Planungswert
 - Die tatsächliche Kosten hängen von vorhandenen Abos, Nutzungsintensität und gegebenenfalls zusätzlicher API- oder Hostingnutzung ab.

In der Veranstaltung kann folgendes Drittanbieter-Tool eingesetzt werden**KI-Tools**

In diesem ausgewählten Seminar der Haufe Akademie spielen KI-Tools eine wichtige Rolle.

Für ein optimales Lernerlebnis ist bei einigen Tools eine Registrierung erforderlich. Die Registrierung ist freiwillig, eine Teilnahme an diesem Seminar ist auch ohne Registrierung möglich. Die Anbieter der Dienste verarbeiten die Daten in der Regel in eigener Verantwortung. Dies bedeutet, dass sie für die Einhaltung der geltenden Datenschutzbestimmungen und die Sicherheit der Verarbeitung verantwortlich sind. Die Haufe Akademie hat keine vertragliche Beziehung zum Anbieter, und die Teilnehmer:innen müssen selbstständig, z.B. Nutzungsbedingungen zustimmen.

Es wird empfohlen, eine E-Mail-Adresse zu verwenden, die keine Rückschlüsse auf eine Person zulässt.

Details zur Verarbeitung sind in der Datenschutzerklärung des jeweiligen Tools wie z.B. [ChatGPT](#), [Gemini](#), [HeyGen](#), [Perplexity](#) etc. zu finden.

Teilnehmer:innenkreis

Das Future Lab AI Coding richtet sich an:

- technisch affine Fachkräfte, sowie an Projekt- und Produktverantwortliche, die einen eigenen B2B-Anwendungsfall mit AI Coding umsetzen wollen.
- Software Entwickler:innen können teilnehmen. Für sie liegt ein Lerngewinn im Loop & Graph Engineering, sowie in systematischen Tests und Reviews (Test Driven Development)

Open Badges - Zeige auch digital, was du kannst.

Open Badges sind anerkannte, digitale Teilnahmezertifikate. Diese verifizierbaren Nachweise sind der aktuelle Standard für die Einbindung in Karrierenetzwerken wie z.B. LinkedIn.

Damit zeigst du digital, über welche Kompetenzen du verfügst.

Nach erfolgreichem Abschluss erhältst du von uns ein Open Badge.

Mehr erfahren kannst du unter:

<https://www.haufe-akademie.de/seminare-lehrgaenge/trending-topics/open-badges>



Details zur Weiterbildung

Seminar | Online

ca. 65 h

max. 12 Teilnehmer:innen

Teilnahmegebühr

€ 2.750,- zzgl. MwSt.

€ 3.272,50 inkl. MwSt.

Haufe Akademie GmbH & Co. KG

Munzinger Straße 9, 79111 Freiburg, www.haufe-akademie.de, Beratung: Tel.: +49 761 595339-00, service@haufe-akademie.de