

VORBEREITUNG AUF UNSEREN TERMIN

Rechner startklar, Material griffbereit.

Damit wir unsere gemeinsame Zeit für dein System nutzen und nicht für Downloads, bereitest du zwei Dinge vor. Du brauchst dafür keine Technikenkenntnisse.

Teil 1 Dein Rechner: vier Programme, rund 30 Minuten.

Teil 2 Dein Material: ein paar Beispiele aus deinem Alltag, rund eine Stunde, ganz ohne Technik.

Was hier passiert und was nicht

Auf dieser Seite bereitest du deinen Rechner und dein Material vor. Das eigentliche System, also dein Wissensspeicher und die Abläufe, die dir täglich Arbeit abnehmen, bringe ich zum Termin mit und richte es dort gemeinsam mit dir auf deinen Betrieb aus. Je besser dein Material vorbereitet ist, desto schneller siehst du das erste echte Ergebnis.

Wenn ein Schritt klemmt: hör einfach auf und mach beim nächsten weiter. Alles, was offen bleibt, erledigen wir zusammen. Das kostet uns zehn Minuten, keine Stunde.

Was du vorher brauchst

- Einen Rechner mit Windows 10 oder 11, oder einen Mac mit macOS 13 (Ventura) oder neuer. Am besten den Rechner, an dem später wirklich gearbeitet wird.
- Deine geschäftliche E-Mail-Adresse.
- Eine Zahlungsmethode für das Abo aus Schritt 1 (Firmenkarte oder Firmenkonto).
- Rund 30 bis 45 Minuten Ruhe für Teil 1 und etwa eine Stunde für Teil 2. Beides muss nicht am selben Tag passieren.

Administratorrechte sind nicht nötig. Falls deine IT Installationen sperrt, sag mir vorher Bescheid, dann klären wir das direkt mit ihr.

TEIL 1: DEIN RECHNER

1 Claude-Konto anlegen und Abo wählen

Das ist der einzige Schritt, der Geld kostet, und deshalb steht er vorne. Claude ist die künstliche Intelligenz, die dein System später bedient. Ohne bezahltes Abo funktioniert es nicht, das kostenlose Konto reicht dafür ausdrücklich nicht.

1. Öffne **claude.ai** und leg ein Konto mit deiner geschäftlichen E-Mail-Adresse an.
2. Wähl unter „Upgrade“ ein Abo. Für den Anfang reicht **Pro**. Wer das System täglich intensiv nutzt, wechselt später auf **Max**.
3. Trag deine Firmendaten als Rechnungsadresse ein, damit die Rechnung sauber in die Buchhaltung geht.

Unsicher, welches Abo? Meld dich kurz bei mir, bevor du abschließt. Ich rechne dir den erwarteten Verbrauch für deinen Fall durch. Ein Wechsel ist jederzeit möglich, du legst dich also nicht fest.

2 Visual Studio Code installieren

Das ist das Fenster, in dem du dein System später bedienst. Es sieht auf den ersten Blick nach Programmierung aus, ist für dich aber vor allem eins: ein Eingabefeld, in das du hineinschreibst, was du brauchst.

| Windows

1. Öffne **code.visualstudio.com** und klick auf die große blaue Schaltfläche „Download for Windows“.
2. Starte die heruntergeladene Datei. Im Installationsfenster: Lizenz annehmen, danach überall auf „Weiter“ und zum Schluss auf „Installieren“. Die Voreinstellungen sind richtig, du musst nichts umstellen.
3. Am Ende „Fertigstellen“ anklicken. Visual Studio Code öffnet sich.
4. Falls oben rechts gefragt wird, ob du ein deutsches Sprachpaket installieren möchtest: Ja, das macht die Bedienung leichter. Das Programm startet danach einmal neu.

| Mac

1. Öffne **code.visualstudio.com** und klick auf die große blaue Schaltfläche „Download for macOS“.
2. Öffne die heruntergeladene Datei im Ordner „Downloads“. Es erscheint ein Fenster mit dem Programmsymbol.
3. Zieh **Visual Studio Code** mit der Maus in den daneben angezeigten Ordner **Programme**. Das ist die Installation, mehr passiert nicht.
4. Öffne das Programm aus dem Ordner „Programme“. Beim ersten Start fragt der Mac, ob du ein aus dem Internet geladenes Programm wirklich öffnen möchtest. Bestätige mit „Öffnen“.
5. Falls oben rechts gefragt wird, ob du ein deutsches Sprachpaket installieren möchtest: Ja, das macht die Bedienung leichter. Das Programm startet danach einmal neu.

Der Startbildschirm mit vielen Kacheln und Links darf dich nicht irritieren. Du brauchst davon nichts. Wir arbeiten beim Termin nur mit einem Ordner und einem Eingabefeld.

3 Claude Code installieren und anmelden

Jetzt kommt die KI auf den Rechner. Hier tippst du einmalig zwei Befehle ab. Das ist der technischste Moment der ganzen Vorbereitung und danach vorbei.

| Windows

1. Öffne in Visual Studio Code oben im Menü **Terminal** und dort **Neues Terminal**. Unten im Fenster erscheint ein Bereich mit heller Schrift auf dunklem Grund, der mit `PS C:\Users\...` beginnt.
2. Schreib dort genau diese Zeile hinein und drück die Eingabetaste:

```
irm https://claude.ai/install.ps1 | iex
```

Der Befehl lädt Claude Code von Anthropic herunter und installiert es. Das dauert ein bis zwei Minuten, dabei laufen Textzeilen durch. Warte, bis wieder eine leere Eingabezeile steht.

3. Schließ Visual Studio Code komplett und öffne es neu. Das ist wichtig, sonst kennt der Rechner den neuen Befehl noch nicht.

| Mac

1. Öffne in Visual Studio Code oben im Menü **Terminal** und dort **Neues Terminal**. Unten im Fenster erscheint ein Bereich mit heller Schrift auf dunklem Grund, dessen Zeile mit einem `%` endet.
2. Schreib dort genau diese Zeile hinein und drück die Eingabetaste:

```
curl -fsSL https://claude.ai/install.sh | bash
```

Der Befehl lädt Claude Code von Anthropic herunter und installiert es. Das dauert ein bis zwei Minuten, dabei laufen Textzeilen durch. Warte, bis wieder eine leere Eingabezeile steht.

3. Schließ Visual Studio Code komplett (im Menü „Visual Studio Code“ auf „Beenden“, nicht nur das Fenster zuklappen) und öffne es neu. Das ist wichtig, sonst kennt der Rechner den neuen Befehl noch nicht.
4. Öffne wieder ein neues Terminal, schreib

```
claude
```

und drück die Eingabetaste. Es öffnet sich dein Browser mit der Anmeldeseite.

5. Melde dich mit dem Konto aus Schritt 1 an und bestätige den Zugriff. Im Browser steht danach, dass du zurück zum Terminal wechseln kannst.
6. Im Terminal begrüßt dich Claude Code. Damit ist es geschafft. Du kannst mit der Eingabe

```
/exit
```

wieder herausgehen.

7. Zum Schluss noch eine kurze Prüfung. Dein System nutzt für ein paar Wartungsroutinen ein weiteres Programm namens **Python**. Auf vielen Rechnern ist es bereits vorhanden, deshalb schauen wir erst nach, bevor du etwas installierst.

| Windows

Schreib ins Terminal

```
py --version
```

Erscheint eine Zeile wie Python 3.12.4, ist alles da. Erscheint eine Fehlermeldung, öffne python.org/downloads, lade die Version für Windows und starte den Installer. Wichtig: Im ersten Fenster unten das Häkchen bei „Add python.exe to PATH“ setzen, bevor du auf „Install Now“ klickst. Das wird leicht übersehen und ist der häufigste Grund, warum es hinterher nicht gefunden wird.

| Mac

Schreib ins Terminal

```
python3 --version
```

Erscheint eine Zeile wie Python 3.12.4, ist alles da und du bist fertig. Erscheint stattdessen eine Meldung über fehlende Entwicklerwerkzeuge, schreib

```
xcode-select --install
```

und bestätige das Fenster, das sich öffnet. Das dauert ein paar Minuten und läuft von allein.

Wenn „claude wird nicht erkannt“ oder „command not found“ erscheint: Das kommt vor und ist harmlos. Das Programm ist installiert, dein Rechner weiß nur noch nicht, wo es liegt. Zwei Dinge lösen das fast immer.

1. Visual Studio Code wirklich ganz schließen und neu öffnen. Am Mac über das Menü „Visual Studio Code“ und dort „Beenden“, nicht nur das Fenster zuklappen.

2. | Mac

Hilft das nicht, tipp diese beiden Zeilen nacheinander ins Terminal, jede mit der Eingabetaste bestätigt:

```
echo 'export PATH="$HOME/.local/bin:$PATH"' >> ~/.zshrc
```

```
source ~/.zshrc
```

Die erste Zeile hinterlegt dauerhaft, wo Claude Code liegt, die zweite übernimmt das sofort. Beide bestätigen nichts sichtbar, das ist richtig so. Prüf danach mit

```
claude --version
```

Erscheint eine Versionsnummer, ist es geschafft.

| Windows

Hilft das nicht, starte den Rechner einmal neu und öffne Visual Studio Code erneut. Prüf dann im Terminal mit

```
claude --version
```

Erscheint eine Versionsnummer, ist es geschafft.

Kommt danach immer noch eine Fehlermeldung, hör hier auf. Wir machen das beim Termin in zwei Minuten zusammen.

Dasselbe gilt für die Python-Prüfung: Klappt sie nicht auf Anhieb, einfach liegen lassen. Der Rest deines Systems funktioniert auch ohne, und wir erledigen es beim Termin in zwei Minuten.

4 Obsidian installieren

Obsidian ist dein Blick ins Gedächtnis des Systems. Alles, was dein System weiß, liegt als ganz normale Textdateien auf deinem Rechner, und Obsidian zeigt sie lesbar an. Es ist kostenlos, auch für die Nutzung im Betrieb.

| Windows

1. Öffne **obsidian.md** und klick auf „Download“ und dann auf die Windows-Version.
2. Starte die heruntergeladene Datei. Obsidian installiert sich ohne Rückfragen und öffnet sich anschließend.

| Mac

1. Öffne **obsidian.md** und klick auf „Download“ und dann auf die Mac-Version (Universal, passt auf jeden Mac).
2. Öffne die heruntergeladene Datei und zieh **Obsidian** in den Ordner **Programme**.
3. Starte Obsidian aus „Programme“ und bestätige die Rückfrage des Macs mit „Öffnen“.

Und jetzt bitte aufhören: Du siehst einen Startbildschirm mit den Auswahlmöglichkeiten „Neuen Tresor erstellen“ und „Ordner als Tresor öffnen“. Klick hier nichts an und schließ das Fenster wieder.

Das ist Absicht: Der Ordner, den Obsidian später öffnen soll, entsteht erst beim Einbau deines Systems. Wenn du jetzt einen leeren Tresor anlegst, haben wir hinterher zwei und müssen aufräumen. Installieren genügt völlig.

Optional: mit der Stimme statt mit der Tastatur

Viele arbeiten deutlich schneller, wenn sie ihre Anliegen einfach sprechen statt tippen. Dein Rechner kann das ab Werk, du musst dafür nichts installieren und nichts bezahlen.

| Windows

Klick in ein Eingabefeld und drück **Windows-Taste + H**. Es erscheint ein kleines Mikrofonfenster. Sprich normal weiter, der Text erscheint dort, wo der Cursor steht. Nochmal Windows-Taste + H beendet die Aufnahme.

| Mac

Schalte das Diktat einmalig ein: **Systemeinstellungen**, dann **Tastatur**, dann **Diktat** auf „Ein“. Danach klickst du in ein Eingabefeld und drückst zweimal kurz die **Steuerungstaste (ctrl)**. Sprich normal weiter, der Text erscheint dort, wo der Cursor steht. Nochmal zweimal ctrl beendet die Aufnahme.

Probier es einmal in einem beliebigen Textfeld aus. Wenn du merkst, dass du viel diktieren, gibt es komfortablere Werkzeuge dafür. Eines davon, Wispr Flow, ist unten im optionalen Teil beschrieben, samt Kosten und der Frage, was dabei wohin übertragen wird.

TEIL 2: DEIN MATERIAL

Jetzt ohne Technik. Dein System lernt nicht aus dem Nichts, sondern aus dem, was du ohnehin schon machen. Je konkreter das Material, desto früher entsteht beim Termin etwas, das dir am nächsten Tag wirklich Arbeit abnimmt. Leg dir dafür auf dem Schreibtisch einen Ordner an, zum Beispiel **Für den Termin**, und sammle dort alles aus den folgenden Schritten.

5 Zwei, drei Zeitfresser aufschreiben

Welche Aufgaben kommen jede Woche wieder und kosten dich mehr Zeit, als sie sollten? Genau dort setzen wir an. Typisch sind Angebote schreiben, Anfragen beantworten, Protokolle und Zusammenfassungen, Texte für Website oder Social Media, Recherche vor einem Kundentermin.

Schreib je Aufgabe zwei, drei Sätze auf, gern stichpunktartig:

- Was ist das Ergebnis, und für wen?
- Wie oft kommt es vor, und wie lange sitzt du jeweils dran?
- Was nervt daran am meisten?

Keine Sorge, wenn dir nur eine Aufgabe einfällt. Eine gut beschriebene ist mehr wert als fünf vage.

6 Beispiele dazulegen

Zu jedem Zeitfresser drei bis fünf echte Beispiele aus den letzten Monaten: frühere Angebote, verschickte Texte, Vorlagen, deine Preisliste oder Kalkulation. Gern auch unordentlich und unfertig, genau so, wie es im Alltag aussieht. Das System liest daraus, wie du arbeitest, und stellt dir dazu Fragen.

Dazu kurz das, was du besonders gut gelungen findest, und das, womit du unzufrieden warst. Beides hilft gleich viel.

7 Angaben zu deinem Betrieb bereitlegen

Beim Termin fragt das System einmal deine Grunddaten ab. Das geht schnell, wenn alles bereitliegt:

- Die Adresse deiner Website, falls es eine gibt. Daraus kann das System vieles vorbefüllen, du bestätigst dann nur noch.
- Firmierung, Anschrift, Telefon, E-Mail und USt-IdNr., wie sie im Impressum stehen.
- Logo, Farben und Schriften, falls du Vorgaben dazu hast. Ein Gestaltungshandbuch ist ideal, eine Logodatei reicht auch.
- Zwei, drei Sätze, die typisch für dich klingen, etwa aus einer Mail an einen guten Kunden.
- Wörter und Floskeln, die du nie schreiben würdest. Das ist der unterschätzteste Punkt der ganzen Liste: Dein System soll nach dir klingen, nicht nach Textbaukasten.

8 Einen echten, aktuellen Fall mitbringen

Eine Anfrage oder Aufgabe, die gerade auf deinem Tisch liegt, in zwei Sätzen: Was will der Kunde, und was musst du dafür liefern? Daraus entsteht beim Termin das erste Ergebnis.

Bitte keinen ausgedachten Testfall. Ob das System dir hilft, siehst du nur an einer Aufgabe, die du diese Woche wirklich erledigen musst.

Wo deine Daten liegen

Damit das von Anfang an klar ist, ohne Kleingedrucktes: Dein Wissensspeicher liegt als gewöhnliche Dateien auf deinem Rechner, nicht bei mir und nicht in einer fremden Datenbank. Du kannst ihn jederzeit kopieren, mitnehmen oder löschen. Auch dein Ordner mit dem Material bleibt bei dir, du musst mir vorab nichts schicken.

Was du das System konkret fragst und was es dafür liest, zum Beispiel deine Beispielangebote, wird zur Beantwortung an Anthropic übertragen, den Anbieter von Claude. Das ist der Teil, der das Haus verlässt, und den erkläre ich dir beim Termin genau. Wenn du bei einem Dokument unsicher bist, lass es bis dahin einfach weg. Wie dein Wissensspeicher gesichert wird, legen wir dann ebenfalls gemeinsam fest, passend zu der Datensicherung, die du ohnehin schon hast.

Kurz zum Abhaken vor dem Termin

- ☐ Claude-Konto angelegt und Abo aktiv
- ☐ Visual Studio Code installiert und einmal geöffnet
- ☐ Claude Code installiert, c1aude startet und ich bin angemeldet
- ☐ Python geprüft (Versionsnummer erscheint) oder für den Termin vorgemerkt
- ☐ Obsidian installiert (noch ohne Tresor)
- ☐ Zeitfresser aufgeschrieben, mindestens einer
- ☐ Beispiele im Ordner „Für den Termin“ gesammelt
- ☐ Website-Adresse, Impressumsangaben und, falls vorhanden, Logo bereitgelegt
- ☐ Einen echten, aktuellen Fall in zwei Sätzen notiert

Gib mir zwei Tage vor dem Termin kurz Bescheid, wie weit du gekommen bist. Ein Foto der abgehakten Liste reicht. Dann weiß ich, ob wir vorab noch etwas klären sollten, und der Termin beginnt mit deinem Betrieb statt mit der Technik.

Vor jedem weiteren Termin

Sobald dein System läuft, ist die Vorbereitung kürzer. Zehn Minuten genügen:

- ☐ **System starten.** In deinem Arbeitsordner /start eingeben. Es erscheint dein Überblick mit den aktuellen Prioritäten.
- ☐ **Sicherung prüfen.** /feierabend läuft am Ende einer Sitzung ohne Fehlermeldung durch.
- ☐ **Aufgabe vom letzten Mal.** Was wir vereinbart haben, ist erledigt oder so weit, wie du gekommen bist.
- ☐ **Ein neuer echter Fall** für das, was wir als Nächstes bauen.

Erscheint irgendwo eine Fehlermeldung: Bildschirmfoto an mich, gern schon ein paar Tage vorher. Lieber vorab eine kurze Nachricht als im Termin eine halbe Stunde Fehlersuche.

OPTIONAL UND TECHNISCHER

Für alle, die mehr wollen

Nichts hier unten brauchst du für den Termin, und nichts davon musst du allein hinbekommen. Wenn ein Punkt zu technisch wirkt, überspring ihn einfach, wir richten das bei Bedarf gemeinsam ein. Ich beschreibe es trotzdem, damit du siehst, was möglich ist und wo deine Daten dabei hingehen.

A Schneller diktieren mit Wispr Flow

Die eingebaute Diktierfunktion weiter oben ist kostenlos und reicht für kurze Sätze. Wer viel spricht statt tippt, kommt mit Wispr Flow spürbar weiter: Du hältst eine Taste, sprichst ganz normal, und der Text landet sauber formatiert dort, wo der Cursor gerade steht. Füllwörter und Satzzeichen setzt das Werkzeug selbst.

| Windows

1. Öffne **wisprflow.ai** und lade die Windows-Version herunter, oder such „Wispr Flow“ im Microsoft Store.
2. Installieren und starten. Das Programm legt sich unten rechts neben die Uhr. Erlaube den Zugriff auf das Mikrofon.
3. Leg ein kostenloses Konto an und melde dich an.
4. Zum Diktieren hältst du **Strg + Windows-Taste** gedrückt und sprichst. Loslassen beendet die Aufnahme.

| Mac

1. Öffne **wisprflow.ai** und lade die Mac-Version herunter.
2. Installieren und starten. Erlaube den Zugriff auf das Mikrofon und auf die Bedienungshilfen. Letzteres ist nötig, damit Flow in andere Programme schreiben darf.
3. Leg ein kostenloses Konto an und melde dich an.
4. Zum Diktieren hältst du die **Fn-Taste** gedrückt und sprichst. Loslassen beendet die Aufnahme.

Kostenlos sind 2.000 Wörter pro Woche, das reicht zum Ausprobieren. Wer dauerhaft viel diktiert, zahlt rund 15 US-Dollar im Monat.

Ehrlich zur Kehrseite

Wispr Flow arbeitet ausschließlich über das Internet, einen Offline-Modus gibt es nicht. Deine Sprache wird auf Servern in den USA in Text umgewandelt, dafür entschlüsselt der Anbieter die Tonaufnahme kurzzeitig. Für normale Notizen ist das unkritisch, bei vertraulichen oder personenbezogenen Inhalten solltest du es abwägen.

Wenn du es nutzt, stell in den Einstellungen den **Privacy Mode** an und den **Cloud Sync** aus. Dann wird nichts fürs Training verwendet und nichts dauerhaft gespeichert. Was das für deine Kundendaten bedeutet, ordnen wir beim Termin gemeinsam ein.

B Sicherung und Gedächtnis an einem zweiten Ort

Dein Wissensspeicher liegt als normale Dateien auf deinem Rechner. Damit nichts verloren geht und du später auch von einem zweiten Gerät darauf zugreifen kannst, sichern wir ihn zusätzlich in einem privaten Online-Speicher. Dafür gibt es zwei etablierte Anbieter. Du legst hier nur das Konto und den Zugangsschlüssel an. **Die eigentliche Verbindung zu deinem System stelle ich nach dem Einbau mit dir zusammen her.**

Welchen der beiden du wählst, ist Geschmackssache, und du kannst dich auch beim Termin entscheiden. Der Unterschied in einem Satz: GitHub ist am weitesten verbreitet und gehört zu Microsoft, GitLab ist unabhängig, quelloffen und lässt sich später sogar ganz auf eigenen Servern betreiben.

Weg A: GitHub

1. Konto anlegen auf github.com mit deiner geschäftlichen E-Mail-Adresse.
2. Oben rechts auf dein Profilbild, dann **Settings**, in der linken Spalte ganz unten **Developer settings**.
3. Dort **Personal access tokens**, dann **Fine-grained tokens**, dann **Generate new token**.
4. Einen Namen vergeben, ein Ablaufdatum wählen und unter **Repository permissions** den Punkt **Contents** auf **Read and write** stellen. Mehr ist nicht nötig.
5. **Generate token** klicken. Der Schlüssel wird nur ein einziges Mal angezeigt. Kopier ihn und leg ihn sicher ab.

Deine Dateien liegen damit auf den Servern von GitHub (Microsoft), überwiegend in den USA.

Weg B: GitLab (unabhängig, quelloffen)

1. Konto anlegen auf gitlab.com.
2. Ein neues Projekt anlegen: **New project**, dann **Create blank project**, einen Namen vergeben und als Sichtbarkeit **Private** wählen.
3. Oben rechts auf dein Profilbild, dann **Edit profile**, in der linken Spalte **Access tokens**.
4. Einen Namen und ein Ablaufdatum vergeben und als Berechtigung **write_repository** anhaken. Das genügt zum Sichern.
5. Token erzeugen. Auch dieser Schlüssel wird nur einmal angezeigt. Kopieren und sicher ablegen.

GitLab gehört keinem großen Konzern, der Kern ist Open Source, und der Speicher lässt sich später auf Wunsch komplett auf eigene Server holen. Das ist der unabhängigere Weg.

Der Zugangsschlüssel ist ein Passwort

Behandle den kopierten Schlüssel wie ein Passwort: nicht per E-Mail oder Chat verschicken, nicht offen ablegen. Am besten notierst du ihn in deinem Passwortmanager. Beim Termin tragen wir ihn gemeinsam an der richtigen Stelle ein.

C Zwei Kürzel für den schnellen Start

Sobald dein System läuft, startest du es täglich. Damit du dafür nicht jedes Mal den vollen Befehl tippen, richten wir zwei kurze Kürzel ein:

- cs** Dein normaler Start jeden Tag: öffnet das System startklar mit deinem Tages-Überblick und arbeitet zügig, ohne bei jedem einzelnen Schritt nachzufragen.
- cc** Dasselbe zügige Arbeiten, aber ohne den Tages-Überblick, wenn du direkt in etwas anderes einsteigen willst.

Wenn du bewusst auf Nummer sicher gehen und die Rückfragen behalten willst, startest du das System stattdessen ohne Kürzel mit dem Befehl `c1aude`.

Am einfachsten lässt du die Einrichtung von Claude Code selbst erledigen. Öffne Claude Code und schreib genau das hinein:

Richte mir zwei dauerhafte Verknüpfungen in meinem Shell-Profil ein. Die Verknüpfung `cs` soll Claude Code mit der Option `--dangerously-skip-permissions` starten und dabei den Befehl `/start` ausführen. Die Verknüpfung `cc` soll Claude Code mit der Option `--dangerously-skip-permissions` starten, ohne einen weiteren Befehl. Erkenne mein Betriebssystem selbst, lege beides dauerhaft an und erkläre mir danach, wie ich die Verknüpfungen benutze und die Eingabe neu lade.

Claude Code fragt dabei einmal nach, ob es deine Profildatei ändern darf. Das ist richtig so, bestätige es. Danach die Eingabe einmal schließen und neu öffnen, dann sind die Kürzel da.

Was „ohne Nachfragen“ bedeutet

Beide Kürzel starten das System im zügigen Modus: Es fragt vor einzelnen Schritten nicht mehr nach, auch nicht, bevor es eine Datei ändert oder löscht. Das ist die Arbeitsweise für den Alltag, wenn du eingearbeitet bist. Dein Sicherheitsnetz ist der normale Befehl `c1aude` ohne Kürzel: Damit fragt das System vor heiklen Schritten wieder nach. Wann du was nimmst, gehen wir beim Termin gemeinsam durch.

Wenn etwas klemmt

Meld dich einfach, auch für Kleinigkeiten. Es ist nichts kaputt, wenn ein Schritt nicht durchläuft, und es ist keine Umstände wert, sich damit lange aufzuhalten.

Erik Schaller, Calyx Advisory

kontakt@calyx-advisory.com

+49 151 505 144 59

Stand: 30. September 2026