



KI ersetzt keine Entwickler – sie verschiebt die Verantwortung.

Warum Unternehmen
jetzt Operatoren
brauchen und
nicht nur Prompts.

von **Matthias Eger**
<https://matthias-eger.de>

Stand:

Weiden, 29.10.2025

Matthias Eger

Pfarräckerstr. 14
92637 Weiden

Tel. +49 (0) 961 634 32 61

Mobil +49 (0) 175 585 73 05

Mail mail@matthiaseger.de

Web www.matthias-eger.de

**Vorwort**

Ich arbeite täglich mit KI. Ich nutze sie aktiv in der Entwicklung. Integriere sie in Prozesse und entwickle sie selbst. Aber ich sehe genauso täglich, wie Unternehmen an KI scheitern. Nicht, weil KI „schlecht“ wäre – sondern weil niemand mehr den Überblick über das große Ganze hat. In diesem Beitrag erkläre ich, warum die Rolle des Operators (DevOp) jetzt wichtiger wird als klassisches „wir lassen was programmieren“, warum die naive Hoffnung „KI baut mir die Software“ brandgefährlich ist, und warum kontrolliertes Informationsmanagement der eigentliche Schlüssel ist.

Abstract / Zusammenfassung zum Mitnehmen

- KI kann Code schreiben. Das ist Fakt.
- KI kann aber kein Projekt führen. Kein Budget schützen. Keine Prioritäten durchsetzen. Keine Haftung übernehmen.
- Ab einer gewissen Komplexität kippt jedes reine „wir fragen halt KI“-Projekt in Chaos. Typische Grenze: mehrere Module, mehrere Beteiligte, mehrere Versionen.
- Genau da kommt die Operator-Rolle ins Spiel: Diese Rolle stellt sicher, dass KI-Ergebnisse konsistent bleiben, dass Architektur-Entscheidungen nicht dauernd umgeworfen werden, dass Verantwortlichkeiten klar sind – und dass am Ende ein Produkt rauskommt, nicht nur Fragment-Code.
- Marmacore® ist dafür mein Werkzeug, um Informationen, Entscheidungen, Anforderungen, Zuständigkeiten, Risiken und tatsächliche Umsetzungsstände kontrolliert zu halten.
- Und weil die Frage am Ende eines jeden Seminars kommt: Ja, man kann mit mir zusammenarbeiten – als Unternehmen, aber auch als Einzelperson mit einer Idee, die Realität werden soll.

1. Der KI-Hype und die Realität, die keiner gern ausspricht

Ich höre zurzeit ständig denselben Satz von Geschäftsführern, Marketingleitern, Gründern und auch einfach kreativen Leuten mit einer guten Idee:

„Brauche ich dich überhaupt noch? Das macht doch jetzt alles die KI.“

Ich verstehe, woher das kommt. Die großen Plattformen zeigen täglich Clips wie:

- „Ich habe mir in 2 Stunden eine komplette App bauen lassen.“
- „Ich habe meine komplette Warenwirtschaft automatisiert.“
- „Ich habe eine Firma ohne Entwickler gegründet, nur mit KI.“

Das sieht glänzend aus. Das klingt nach: „Softwareentwicklung ist tot, weil KI Code schreibt.“ Und ja, KI schreibt heute schon erstaunlich guten Code. Ich sage das ganz offen: Ich selbst nutze KI in meiner täglichen Arbeit. Ich lasse mir Code generieren. Ich lasse mir Varianten berechnen. Ich lasse mir auch Dinge erklären, die ich schneller erklärt haben möchte, als sie in der Doku zu suchen.

Der Punkt ist: **Das ist nicht das Problem.**

Das Problem kommt später.

Das Problem kommt nicht beim ersten Snippet.

Das Problem kommt nicht bei „mach mir bitte ein Login-Formular“.

Das Problem kommt nicht bei „bau mir eine Seite, die ein PDF generiert“.

Das Problem kommt dann, wenn aus einem Einzelschnipsel ein System wird.

Spätestens wenn mehrere Module zusammenspielen müssen – Zahlungslogik, Benutzerverwaltung, Sicherheitsprüfung, Reporting, Automatisierung, Rechtekonzepte, Logs, Backups, Updates, Wartbarkeit, Performance unter Last, Datenschutz, Haftung – dann endet das Versprechen „KI macht das für dich“ ziemlich abrupt.

Ich sage das bewusst so klar, weil das der entscheidende Schmerzpunkt ist, den dir die ganzen KI-Showcases nicht erzählen:

KI kann dir problemlos 1.000 Zeilen Code generieren. Aber KI übernimmt keine Verantwortung dafür, dass diese 1.000 Zeilen Code morgen noch funktionieren, wenn du das System weiter ausbaust.

Und genau das ist der Unterschied zwischen „wir haben uns was mit KI bauen lassen“ und „wir haben ein Produkt, das wir wirklich betreiben können“.

2. Warum KI in echten Projekten an Grenzen stößt

Ich beschreibe dir jetzt Situationen, die ich inzwischen ständig sehe. Vielleicht erkennst du dich in einer davon wieder. Wenn ja: Das ist normal. Du bist nicht „zu blöd für KI“. Das ist Architektur.

2.1 Das Kontextproblem: KI vergisst

KI arbeitet im Chat. Ein Chat hat Fenstergröße. Du fragst etwas, bekommst Antwort. Beim nächsten Mal fragst du etwas anderes, bekommst Antwort. Für kleine Aufgaben ist das genial.

Aber:

Wenn du ein Softwareprojekt hast mit vielen Modulen, vielen gegenseitigen Abhängigkeiten, dann zählt jede einzelne Annahme.

- „Wir speichern Kundendaten nicht in Klartext.“
- „Die Rolle ‚Mitarbeiter intern‘ darf Bestellungen sehen, aber nicht exportieren.“
- „Feld A darf nur befüllt sein, wenn Status B auf ‚frei gegeben‘ steht.“
- „Diese Logik muss DSGVO-konform sein, weil dort personenbezogene Daten ins Reporting laufen.“

Das sind alles Entscheidungen. Das sind Regeln. Das sind Grenzen. Das sind Versprechen, die du deinem eigenen Unternehmen gegenüber abgibst.

Jetzt passiert Folgendes:

- Du arbeitest weiter.
- Du fragst die KI später etwas anderes.
- Die KI „sieht“ diese alten Entscheidungen nicht mehr (weil der Chat-Kontext nicht unendlich groß ist).
- Die KI ändert stillschweigend die Architektur.

Das merkst du nicht sofort. Erst Wochen später, wenn plötzlich Kunden X auf Daten zugreifen kann, die er nie sehen dürfte. Oder wenn ein Modul erstaunlich langsam ist, weil zwei unterschiedliche Datenmodelle parallel existieren, die nie zusammengeführt wurden. Oder wenn ein Export plötzlich personenbezogene Daten rausschiebt, die rechtlich gar nicht rausdürfen.

Ich habe das live gesehen. Mehrfach.

Das ist kein exotisches Randproblem.
Das ist Alltag.

Und es ist nicht die „Schuld der KI“.

Es ist ein strukturelles Limit: reine Chat-KI ohne Projektorchestrierung kann keinen stabilen, historischen Wissensstand halten, der als Verbindlichkeit über dem gesamten Projekt liegt.

Ich formuliere das mal übersetzt ins echte Leben:

KI ist wie ein extrem kluger Praktikant, der dir heute extrem sauber etwas baut, aber sich morgen null daran erinnert, was gestern beschlossen wurde – und deswegen die halbe Firma umbaut, ohne Rücksprache.

Wenn du lachst: das ist genau das Level an Risiko, das unterschätzt wird.

2.2 Das Integrationsproblem: Die Bausteine passen nicht zusammen

Viele Leute machen Folgendes (verständlicherweise):

1. Sie lassen sich ein Modul A bauen (z. B. Rechnungs-PDF-Generator).

2. Ein paar Tage später lassen sie sich ein Modul B bauen (z. B. Kundenverwaltung mit Mahnstufen).
3. Dann kommt Modul C (Reporting für Buchhaltung).
4. Später Modul D (Dashboard fürs Management mit KPIs).

Klingt super: Viermal KI gefragt, viermal generiert, viermal Erfolg.

Bis zu dem Moment, wo Modul B etwas über Kunden wissen möchte, was nur in Modul A gepflegt wird.

Plötzlich muss das System miteinander reden.

Und jetzt knallt es.

Warum?

Weil Modul A vielleicht in PHP entstanden ist, Modul B in Node.js, Modul C in einer Excel-Export-Logik lebt und Modul D halb in irgendeinem Low-Code-Baukasten hängt und halb Custom-JavaScript ist, das irgendwer nachts um 1 Uhr generieren ließ.

Aus KI-Sicht ist jede einzelne Antwort „richtig“.

Aus Projektsicht hast du jetzt ein Frankenstein-System, das keiner mehr kontrolliert.

Ich habe Projekte gesehen, in denen fünf Programmiersprachen parallel laufen – nicht aus architektonischer Absicht, sondern weil man sich schrittweise Dinge von der KI hat generieren lassen, ohne jemanden drüberzusetzen, der sagt:

„Stopp. Wir entscheiden uns jetzt für EINEN Stack, wir definieren eine Schnittstelle, wir bauen eine saubere Datenstruktur. Und zwar verbindlich.“

Ohne diese Klammerfunktion eskalieren die Wartungskosten nach hinten.

Du sparst am Anfang 5.000 €.

Du verbrennst hinten 50.000 €.

2.3 Das Verantwortungsproblem: Wer haftet?

Ein Punkt, über den kaum jemand spricht:

Wenn du dir Code von KI generieren lässt und den live auf Kunden loslässt – wer trägt die Verantwortung?

- Wenn personenbezogene Daten offenliegen.
- Wenn fälschbar Rechnungen generiert werden.
- Wenn jemand unberechtigt Einblick in interne Abläufe bekommt.
- Wenn etwas „versehentlich“ Rückschlüsse auf Preise, Margen, Lieferanten erlaubt.

Die KI? Nein.

Der Prompt-Schreiber? Formal schon.

Und wer ist der Prompt-Schreiber? Am Ende du.

Das ist also nicht „nur ein bisschen Software“.

Das ist geschäftskritisch.

Und das ist einer der Gründe, warum ich sage:
Es ist gefährlich, KI als „Selbstbedienungs-Softwarefabrik“ zu verkaufen.
Weil die Fabrik keine Haftung übernimmt.
Du schon.

3. Warum wir jetzt eine neue Rolle brauchen: den Operator

Ich nenne diese Rolle ganz bewusst „Operator“ bzw. „DevOp“ (Development + Operation, also Entwicklung + Betrieb).

Was macht ein Operator?

Ein Operator führt KI, statt sich von KI führen zu lassen.

Ein Operator:

- hält die Architektur zusammen,
- sorgt dafür, dass die getroffenen Entscheidungen konsistent bleiben,
- stellt sicher, dass Module zueinander passen,
- erkennt früh, wann KI Unsinn baut,
- priorisiert,
- dokumentiert,
- entscheidet, welche Lösung produktiv gehen darf und welche nicht.

Das ist kein klassischer „Fullstack-Entwickler“ mehr im alten Sinn nach dem Motto „ich tippe Code“.

Das ist eher so etwas wie Regisseur + Sicherheitsbeauftragter + Architekt + Integrator.

Ich formuliere es aus meiner Sicht:

Meine Aufgabe ist nicht mehr, jeden einzelnen Controller von Hand zu tippen.

Meine Aufgabe ist, dafür zu sorgen, dass am Ende ein System existiert, das du wirklich betreiben kannst – ohne dass dir das Ding nach drei Monaten explodiert.

Ich kann KI benutzen, um schneller zu bauen. Natürlich.

Ich kann KI benutzen, um Varianten zu bekommen. Natürlich.

Ich kann KI auch benutzen, um in bestehendem Code Dinge zu verbessern oder zu erklären.
Natürlich.

Aber ich muss die Leitplanken setzen:

- Was darf sich ändern?
- Was darf sich NIE ändern?
- Wo liegen die sensiblen Daten?
- Welche Teile müssen versioniert werden?
- Wer darf wann wohin deployen (also live schalten)?
- Gibt es Backups?

- Gibt es eine saubere Übergabe?

Diese Rolle ist kein Luxus mehr.

Diese Rolle ist Überlebenssicherung.

Weil KI zwar schnell baut – aber genauso schnell kaputtbauen kann.

4. Wie sich Softwareentwicklung gerade wirklich verändert

Früher war Softwareentwicklung oft:

„Setz dich hin, schreib das Modul, teste, fertig.“

Heute ist Softwareentwicklung eher:

„Wir haben 8 Bausteine, von denen 3 KI-generiert sind, 2 kommen aus vorhandenen Altsystemen, 2 sind extern (Zahlungsanbieter, CRM) und 1 ist komplett neu. Bring das bitte so zusammen, dass es:

- stabil läuft,
- sicher ist,
- wartbar bleibt,
- und nicht morgen doppelt so teuer wird wie heute.“

Das heißt:

Die eigentliche Arbeit verschiebt sich weg vom Coden einzelner Funktionen hin zum Zusammenführen, Absichern, Strukturieren, Betreiben.

Das ist übrigens der Moment, wo viele kleine und mittelständische Unternehmen komplett aussteigen. Weil das extrem unangenehm ist. Das ist der Bereich, in dem „mal schnell ausprobieren“ nicht mehr reicht.

Und jetzt kommt der Punkt, den kaum jemand offen anspricht:

Ganz viele Leute haben es bereits versucht.

Sie haben wirklich ernsthaft versucht, sich mit KI eine kleine ERP-Lösung zu bauen, eine Zeiterfassung, eine Ressourcenplanung, ein Ticketsystem, ein Bestellportal, eine Angebotsstrecke, eine Lead-Qualifizierung, you name it.

Ich spreche hier nicht über Theoretiker.

Ich spreche über Menschen, die nachts um halb eins mit ChatGPT ringen, weil am nächsten Morgen irgendwas funktionieren muss.

Und in ganz vielen Fällen passiert das Gleiche:

- die Idee war gut,
- die ersten Schritte liefen großartig,
- die Begeisterung war riesig,
- dann kam die erste Abzweigung,
- dann das erste echte Problem mit echtem Datenfluss,

- dann die erste Sicherheitsfrage,
- dann hat das Kartenhaus gewackelt,
- und dann wurde das Projekt still begraben.

Nicht wegen fehlender Intelligenz.

Nicht wegen Faulheit.

Sondern weil niemand die Rolle übernommen hat, das Ganze auf Schienen zu setzen.

Das ist die neue harte Wahrheit:

Die Fähigkeit, Bausteine zu orchestrieren, ist heute wertvoller als die Fähigkeit, einzelne Bausteine zu schreiben.

5. Wo KI den Menschen niemals ersetzen wird

Ich möchte an der Stelle bewusst klar sein, weil das gern weichgespült wird:

Es gibt Bereiche, in denen KI gut ist. Und es gibt Bereiche, in denen KI sehr, sehr schlecht ist.

5.1 Prioritäten setzen

KI weiß nicht, was für dein Unternehmen wirklich wichtig ist.

Ein Beispiel:

- Du fragst: „Baue mir bitte eine Lagerverwaltung.“
- Die KI sagt: „Hier ist eine Lagerverwaltung.“
- Sie baut dir sauber Tabellen, Felder, Logik.

Aber:

Die KI weiß nicht, dass du rechtlich verpflichtet bist, bestimmte Chargen-Rückverfolgbarkeit zu gewährleisten.

Sie weiß nicht, dass dein größtes Risiko nicht Lager X ist, sondern Rückruf Y.

Sie weiß nicht, dass ein Verlust von 10 Artikeln A schlimmer ist als ein Verlust von 300 Artikeln B.

Prioritätensetzung ist Strategie. Strategie kommt aus Einblick.

Das ist nicht stumpf ableitbar aus Code.

5.2 Verantwortung übernehmen

Eine KI wird dir niemals sagen:

„Ich gehe für diese Architektur persönlich gerade mit meinem Namen ins Risiko.“

Ich schon.

Und das ist nicht als Show gemeint. Das ist ernst.

Wenn ich sage: „Wir können das so machen“, dann meine ich:

- Ich halte das für tragbar.
- Ich halte das für technisch sinnvoll.

- Ich halte das für wirtschaftlich vertretbar.
- Ich halte das für sicher genug vor Kundendaten.
- Ich halte das für wartbar in 6 Monaten ohne kompletten Rewrite.

Ich verkaufe dir kein Zauberwort.

Ich gehe mit dir in die Verantwortung, dass das Ding dann auch noch läuft, wenn der erste Hype wieder vorbei ist.

5.3 Menschen verstehen

KI kann keinen Menschen beruhigen, der nervös ist, weil er glaubt, seine ganze Firma sei plötzlich „zu langsam“ oder „zu alt“.

KI kann nicht entscheiden, wo eure Kapazität wirklich hin muss: in Marketing? in Automatisierung? in Angebotsstrecke? in Lieferkette?

Dieser Blick ist menschlich-politisch.

Und das ist kein Fehler. Das ist gesund.

6. Marmacore® – warum ich damit arbeite

Jetzt kommen wir zu meinem Werkzeug.

Ich habe Marmacore® (<https://marmacore.software>) aufgebaut, weil ich irgendwann gemerkt habe:

Das eigentliche Problem ist nicht „wir brauchen mehr Code“.

Das eigentliche Problem ist „wir verlieren den Überblick über unser eigenes System“.

Marmacore® ist im Kern ein Informationsmanagement-System.

Was heißt das praktisch?

6.1 Entscheidungen werden festgehalten

Jede wichtige Entscheidung im Projekt wird sauber erfasst:

- Warum haben wir uns für genau diese Datenstruktur entschieden?
- Welche Felder dürfen niemals rausgegeben werden?
- Welche Schnittstelle ist verbindlich?
- Welche Version ist live? Welche ist Test?
- Was ist „muss“, was ist „nice to have“?

Warum ist das wichtig?

Weil KI diesen Langzeit-Kontext nicht halten kann.

Marmacore® schon.

Ich zwingen die Projektlogik in eine Form, die langfristig nachvollziehbar ist.
Wenn später etwas angepasst werden soll, dann passiert das auf Basis von dokumentierten Entscheidungen – nicht aus dem Bauch der KI heraus.

6.2 Wissen ist nicht mehr verteilt in Köpfen und Chats

Normalerweise läuft es so:

- Du hast Chatverläufe.
- Du hast E-Mails.
- Du hast Sprachnachrichten.
- Du hast Task-Listen.
- Du hast einzelne Dateien.
- Du hast Halbwissen im Kopf einer einzelnen Person.

Und dann geht diese Person in Urlaub. Oder wechselt. Oder ist krank.
Und dann brennt's.

Magmacore® führt diese Fäden zusammen:

- Anforderungen
- Einschränkungen
- Risiken
- Stände der Module
- wer an welchem Teil „drehen darf“
- was bereits live ist (und bitte nicht mehr ungefragt angefasst wird)

Das klingt banal, ist aber brutal wichtig in einer Welt, in der KI ständig neue Vorschläge rausschießt. Du brauchst eine Instanz, die sagt: „Nein. Das da bleibt so. Punkt.“

6.3 Geschwindigkeit ohne Chaos

Ich biete über Magmacore® eine klare Zusage an:

- ab 149 € pro Monat bekommst du strukturierte Unterstützung,
- Probleme werden innerhalb von 24 Stunden adressiert.

Was heißt „adressiert“?

Nicht: „Wir versprechen dir jedes Mal die ultimative Komplettlösung in 24 Stunden.“ Das wäre Quatsch.

Sondern: „Das Thema landet nicht mehr im Nirvana. Es wird eingeordnet, bewertet, priorisiert, und in einen sauberen nächsten Schritt überführt.“

Für viele kleine und mittlere Unternehmen ist genau das der Unterschied zwischen „wir verzetteln uns schon wieder, lassen wir das Projekt einfach sterben“ und „wir kommen endlich vorwärts“.

Und ich sage dir auch offen:

Magmacore® ist nicht einfach ein weiteres Tool unter vielen.

Magmacore® ist eine Arbeitsweise.

Es zwingt dazu, Entscheidungen bewusst zu treffen, statt sie der KI zu überlassen, nur weil grad keiner hinschaut.

6.4 Magmacore® ist kein Baukasten, der dich ersetzt

Das ist mir wichtig.

Magmacore® ist nicht: „Klick dir deine Firma zusammen, und brauchst nie wieder jemanden.“

Ich verspreche dir nicht „kein Entwickler mehr nötig“.

Ich verspreche dir „kein Blindflug mehr“.

Magmacore® stellt sicher, dass:

- wir KI gezielt einsetzen, wo KI stark ist,
- wir menschliche Entscheidungen festhalten, wo Verantwortung getragen werden muss,
- und wir schnell reagieren können, wenn etwas brenzlig wird.

Das ist der Punkt, wo es professionell wird.

7. Coaching: Für Menschen mit Vision (nicht nur Firmen)

Das hier ist mir persönlich wichtig, und ich sage es sehr direkt:

Ich will nicht nur für Unternehmen arbeiten.

Ich will auch für Einzelpersonen arbeiten.

Ich meine genau dich, wenn du gerade liest und denkst:

- „Ich hab da eine Idee.“
- „Ich sehe da echt eine Marktlücke.“
- „Ich glaube, das wäre verkaufbar, wenn ich es technisch hinbekomme.“
- „Ich weiß nur nicht, wo ich anfangen soll, weil ich kein Entwickler bin.“

Ganz ehrlich:

Das ist kein Grund mehr, heute nicht zu starten.

Aber (und das ist ein dickes Aber):

Du brauchst jemanden, der dafür sorgt, dass du dich nicht sofort verrennst.

Ich biete Coaching an für genau diese Situation:

- Du musst kein Hardcore-Entwickler werden.

- Du musst nicht jede Codezeile verstehen.
- Du musst nicht wissen, wie man eine API technisch bis zur letzten Kante absichert.

Was du verstehen musst, ist:

- Wie du mit KI so arbeitest, dass die KI dir dient statt dich zu verwirren.
- Wie du dein Projekt so beschreibst, dass es baubar wird.
- Wie du erkennst, wann du in echtes Risiko läufst (rechtlich, technisch, finanziell).
- Und wie du deine Idee in etwas verwandelst, das nicht nur ein Traum bleibt, sondern ein Produkt wird.

Ich helfe dir, diese Lücke zu schließen.

Ich helfe dir, nicht an denselben Stellen zu scheitern, an denen schon tausend Leute vor dir gescheitert sind:

- „Ich hab’s mit KI versucht, es war am Anfang so geil, und dann ist alles auseinandergefallen.“
- „Ich hab ein halbfertiges Tool ohne Login-Absicherung, und ich hab Angst, es jemandem zu zeigen.“
- „Ich hab keinen Überblick mehr, was jetzt aktuell ist und was alt ist.“
- „Ich weiß nicht mehr, ob mein Stand rechtlich überhaupt ok ist.“

Wenn du dich da erkennst: Das ist normal. Das ist reparierbar.

Das ist genau der Moment, wo es Sinn macht, mich reinzuholen.

8. Fazit: Die Zukunft gehört nicht denen, die am lautesten „KI!“ schreien. Sondern denen, die führen.

Ich sage es so deutlich, wie ich es denke:

KI hat die Spielregeln verändert.

Die klassische Beauftragung „Programmier mir was, und dann läuft das zehn Jahre“ ist tot.

Genauso tot ist aber auch die naive Hoffnung „Ich frag einfach KI und dann hab ich ein fertiges Produkt“.

Die Zukunft liegt dazwischen.

Die Zukunft gehört den Menschen und Unternehmen, die Folgendes ernst nehmen:

1. Wir nutzen KI offensiv, ja. Weil sie Geschwindigkeit gibt.
2. Wir akzeptieren aber, dass KI kein Verantwortlicher ist.
3. Wir bauen Systeme, keine Code-Haufen.
4. Wir halten Wissen zentral – nicht verteilt auf zufällige Chatlogs.
5. Wir entscheiden bewusst, was live darf und was nie live darf.
6. Wir verhindern Chaos früh, statt es später teuer aufzuräumen.

Das ist Operator-Arbeit.
Das ist DevOp.
Das ist mein Job.

Und Marmacore® ist die Plattform, mit der ich diese Arbeit strukturiert mit dir umsetze – egal ob du ein Mittelständler bist, der gerade versucht, interne Abläufe zu digitalisieren, oder eine einzelne Person mit einer richtig guten Idee, die einfach niemand sonst ernst nimmt.

Ich stehe dazu. Mit meinem Namen.
Und ich übernehme Verantwortung.

Wenn du an dem Punkt bist, wo du sagst:

- „Wir haben intern schon was gebastelt, aber das ist ehrlich gesagt instabil.“
- oder „Wir glauben an unsere Idee, aber wir kommen technisch nicht durch.“
- oder „Wir sind mitten in KI-Chaos und brauchen Übersicht, keine Versprechen.“

Dann bist du genau die Art Mensch, für die ich das hier schreibe.

Bereit, KI richtig einzusetzen – statt dich von ihr überrollen zu lassen?

Wenn du diesen Text bis hierhin gelesen hast, dann hast du vermutlich erkannt, dass der Unterschied zwischen „KI nutzen“ und „KI beherrschen“ riesig ist. Genau da setze ich an.

Ich helfe dir, aus Chaos wieder Kontrolle zu machen –
ob als Unternehmen oder als Einzelperson mit einer Idee, die einfach funktionieren soll.
Gemeinsam bringen wir Struktur in dein Projekt, schaffen Klarheit über nächste Schritte und sorgen dafür, dass KI für dich arbeitet – nicht gegen dich.

 **Melde dich jetzt unverbindlich über mein Kontaktformular:**

 **Kontakt aufnehmen**

Oder direkt über <https://marmacore.software>, wenn du wissen willst, wie Marmacore® dein Informationsmanagement auf das nächste Level bringt.

Ich freue mich, von dir zu hören –

Matthias Eger

Operator. Entwickler. Möglichmacher.

Matthias Eger Design Studio

Inhaber Matthias Eger
Pfarräckerstr. 14
92637 Weiden

Tel. +49 (0) 961 634 32 61

Mobil +49 (0) 175 585 73 05

Mail mail@matthiaseger.de

Web www.matthias-eger.de

