

PROMPT ENGINEERING MEISTERN

BAND 2

Prompt-Frameworks

Strukturiert zum perfekten Prompt

Belkis Aslani

2026

Prompt Engineering Meistern

Band 2: Prompt-Frameworks – Strukturiert zum perfekten Prompt

© 2026 Belkis Aslani. Alle Rechte vorbehalten.

1. Auflage, März 2026

Dieses Werk ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die in diesem Buch genannten Produkt- und Firmennamen sind Marken der jeweiligen Eigentümer.

Satz und Layout: Eigensatz des Autors

Umschlaggestaltung: Belkis Aslani

Inhaltsverzeichnis

Vorwort

- 1 Warum Frameworks?
- 2 Zero-Shot Prompting – Ohne Beispiel zum Ziel
- 3 One-Shot Prompting – Ein Beispiel sagt mehr als tausend Worte
- 4 Few-Shot Prompting – Muster erkennen durch mehrere Beispiele
- 5 Das CRAFT-Framework – Dein Allrounder
- 6 Das RTF-Framework – Schnell und effektiv
- 7 Das RISEN-Framework – Für die komplexen Fälle
- 8 Frameworks im Vergleich – Welches wann?
- 9 Deine persönliche Template-Bibliothek
- 10 Zusammenfassung und Ausblick

Vorwort

Willkommen zurück.

Wenn du Band 1 gelesen hast, weißt du jetzt, was KI ist, wie LLMs funktionieren und wie du grundlegende Prompts schreibst. Du kennst die 5 Bausteine, du weißt, warum Kontext wichtig ist, und du hast hoffentlich dein erstes Prompt-Protokoll angefangen.

Aber ich wette, du hast auch gemerkt: Manchmal sitzt du vor dem Eingabefeld und weißt nicht, wo du anfangen sollst. Du hast eine vage Idee, was du willst, aber der Weg von der Idee zum fertigen Prompt fühlt sich an wie Improvisieren.

Genau da setzt Band 2 an.

Was sich jetzt ändert

In Band 1 hast du gelernt, *was* ein guter Prompt braucht. In Band 2 lernst du, *wie* du ihn systematisch aufbaust. Der Unterschied? Stell dir vor, du kochst. Band 1 hat dir beigebracht, welche Zutaten gut zusammenpassen. Band 2 gibt dir jetzt die Rezepte.

Diese Rezepte heißen Frameworks. Und bevor du denkst “Oh nein, schon wieder so ein Buzzword” – lass mich erklären, warum Frameworks keine Theorie-Spielerei sind, sondern echte Zeitersparnis.

Ich habe Monate gebraucht, um meinen eigenen Workflow zu finden. Ich habe tausende Prompts geschrieben, die meisten davon mittelmäßig. Irgendwann habe ich angefangen, Muster zu erkennen. Die besten Prompts hatten alle eine ähnliche Struktur. Und als ich diese Struktur bewusst anwendete, wurden meine Ergebnisse sofort besser.

Diese Strukturen – CRAFT, RTF, RISEN – teile ich in diesem Band mit dir.

Was dich erwartet

Drei große Themenblöcke:

Block 1: Die Shot-Typen (Kapitel 1–3)

Zero-Shot, One-Shot, Few-Shot – klingt nach Kaffeebestellungen, ist aber die Grundlage für alles, was kommt. Du lernst, wann du dem Modell Beispiele gibst und wann nicht.

Block 2: Die Frameworks (Kapitel 4–7)

CRAFT, RTF und RISEN im Detail. Jedes Framework mit Erklärung, Beispielen und Übungen. Plus ein Vergleich, damit du weißt, wann du welches nutzt.

Block 3: Deine Template-Bibliothek (Kapitel 8–9)

Du baust dir eine persönliche Sammlung von Prompt-Vorlagen auf, die du immer wieder verwenden kannst. Wie ein Koch sein Rezeptbuch – nur für KI-Kommunikation.

Wie du dieses Buch nutzt

Wie schon in Band 1: Lies es der Reihe nach. Die Kapitel bauen aufeinander auf. Und mach die Übungen. Ich kann das nicht oft genug sagen. Du lernst Prompting nicht durch Lesen. Du lernst es durch Machen.

Noch was: Ab diesem Band wirst du merken, dass die Prompts länger werden. Das ist normal. Und es ist gut. Längere Prompts sind nicht automatisch bessere Prompts – aber strukturierte Prompts sind fast immer besser als unstrukturierte. Und Struktur braucht Platz.

Also: Klapp den Laptop auf, öffne dein LLM deiner Wahl, und lass uns loslegen.

Belkis Aslani, März 2026

Kapitel 1: Warum Frameworks?

Bevor wir in die einzelnen Frameworks eintauchen, müssen wir eine grundlegende Frage klären: Warum braucht man überhaupt Frameworks für Prompts?

Die ehrliche Antwort: Brauchst du nicht. Du kannst auch ohne Frameworks gute Prompts schreiben. Du hast die 5 Bausteine aus Band 1, du hast Erfahrung gesammelt, und manchmal reicht ein einfacher Satz, um genau das Ergebnis zu bekommen, das du willst.

Aber.

Das Problem mit der Intuition

Kennst du das? Du schreibst einen Prompt, bekommst ein brauchbares Ergebnis und denkst dir: "Läuft." Dann, drei Tage später, schreibst du einen ähnlichen Prompt für eine ähnliche Aufgabe – und das Ergebnis ist Müll. Du weißt nicht, was du beim ersten Mal anders gemacht hast. Du erinnerst dich nicht an die genaue Formulierung. Und dein Prompt-Protokoll? Hast du vielleicht doch nicht so konsequent geführt.

Das ist das Problem mit Intuition. Sie funktioniert – manchmal. Aber sie ist nicht reproduzierbar. Und in dem Moment, wo du KI regelmäßig nutzt, brauchst du Reproduzierbarkeit.

Frameworks lösen genau dieses Problem.

Was ein Framework tut

Ein Prompt-Framework ist nichts anderes als eine Checkliste. Eine Vorlage, die dir sagt: “Denk an diesen Punkt. Und an diesen. Und an diesen.” Es zwingt dich, bestimmte Elemente in deinen Prompt einzubauen, die du sonst vielleicht vergessen würdest.

Das klingt banal. Ist es auch. Aber die banalsten Dinge sind oft die effektivsten.

Piloten benutzen Checklisten. Nicht weil sie dumm sind, sondern weil Checklisten Fehler verhindern. Chirurgen benutzen Checklisten. Nicht weil sie ihr Handwerk nicht können, sondern weil in Stresssituationen selbst Profis Dinge vergessen.

Prompt-Frameworks sind deine Checklisten.

Die drei Ebenen

Es gibt verschiedene Ebenen, auf denen Frameworks arbeiten:

Ebene 1: Shot-Typen

Die grundlegendste Ebene. Hier geht es um eine einzige Frage: Gibst du dem Modell Beispiele oder nicht?

- **Zero-Shot:** Keine Beispiele. Du beschreibst nur die Aufgabe.
- **One-Shot:** Ein Beispiel.
- **Few-Shot:** Mehrere Beispiele.

Das klingt simpel, und das ist es auch. Aber die Wirkung ist enorm. Wir behandeln das ausführlich in den Kapiteln 2–4.

Ebene 2: Struktur-Frameworks

Hier wird es interessanter. Struktur-Frameworks geben dir eine feste Reihenfolge vor, in der du deinen Prompt aufbaust:

- **CRAFT:** Context, Role, Action, Format, Tone
- **RTF:** Role, Task, Format
- **RISEN:** Role, Instructions, Steps, End goal, Narrowing

Jedes Framework hat seine Stärken. CRAFT ist am universellsten, RTF ist am schnellsten, RISEN ist am detailliertesten. In den Kapiteln 5–8 lernst du alle drei kennen und erfährst, wann du welches nutzen solltest.

Ebene 3: Template-Bibliothek

Die dritte Ebene ist deine persönliche Sammlung. Du nimmst die Frameworks, wendest sie auf wiederkehrende Aufgaben an und speicherst die besten Prompts als Vorlagen. Kapitel 9 zeigt dir, wie du das aufbaust.

Frameworks sind keine Zwangsjacke

Ich muss das betonen, weil ich es in jeder Diskussion über Frameworks höre: “Aber dann wird doch alles so starr und formelhaft!”

Nein. Wird es nicht.

Ein Framework ist ein Startpunkt, kein Korsett. Du benutzt es als Grundlage und passt es an deine Bedürfnisse an. Manchmal lässt du Teile weg. Manchmal fügst du Elemente hinzu. Das ist nicht nur erlaubt, das ist gewünscht.

Denk an Musik. Professionelle Musiker lernen jahrelang Musiktheorie. Tonleitern, Harmonielehre, Rhythmik. Macht sie das steif und formelhaft? Nein. Es gibt ihnen das Fundament, auf dem sie improvisieren können.

Prompt-Frameworks funktionieren genauso. Sie geben dir Struktur, damit deine Kreativität nicht ins Leere läuft.

Der Vorher-Nachher-Effekt

Lass mich dir ein konkretes Beispiel zeigen.

Ohne Framework:

Schreib mir einen Blogartikel über gesunde Ernährung.

Das Ergebnis? Generisch. Langweilig. Austauschbar. Könnte von jedem geschrieben worden sein, für niemand bestimmtes.

Mit CRAFT-Framework:

Context: Ich betreibe einen Blog über alltagstaugliche Gesundheitstipps für berufstätige Eltern. Meine Leser haben wenig Zeit und wollen praktische, sofort umsetzbare Ratschläge.

Role: Du bist ein erfahrener Ernährungsberater, der sich auf Familienernährung spezialisiert hat. Du kommunizierst locker und praxisnah, ohne erhobenen Zeigefinger.

Action: Schreibe einen Blogartikel über 5 einfache Wege, wie berufstätige Eltern die Ernährung ihrer Familie verbessern können, ohne stundenlang in der Küche zu stehen.

Format: 800-1000 Wörter. Einleitung, 5 nummerierte Tipps mit jeweils einer konkreten Umsetzungsidee, Fazit. Verwende Zwischenüberschriften.

Tone: Locker, ermutigend, realistisch. Kein Schuldgefühlmachen. Kein "Du musst" – sondern "Du kannst".

Der Unterschied im Ergebnis ist dramatisch. Und das Beste: Du musst nicht jedes Mal von Null anfangen. Wenn du nächste Woche einen weiteren Blogartikel brauchst, nimmst du diesen Prompt, änderst das Thema, und fertig.

Das ist die Kraft von Frameworks.

Ein Wort der Warnung

Frameworks machen dich nicht automatisch besser. Sie sind Werkzeuge, keine Wundermittel. Wenn dein Verständnis der Grundlagen fehlt – wenn du nicht weißt, warum Kontext wichtig ist oder wie Rollen funktionieren –, dann helfen dir auch die schicksten Frameworks nicht weiter.

Deshalb war Band 1 so wichtig. Und deshalb sage ich: Wenn du Band 1 übersprungen hast, geh zurück und lies ihn. Die Frameworks in diesem Band setzen voraus, dass du die Grundlagen draufhast.

Für alle anderen: Auf geht's.

Übung

Framework-Reflexion

Nimm drei Prompts, die du in den letzten Tagen geschrieben hast (oder schreib drei neue für alltägliche Aufgaben). Für jeden Prompt:

1. Identifiziere, welche der 5 Bausteine (Aufgabe, Kontext, Format, Ton, Einschränkungen) du verwendet hast
2. Notiere, welche du vergessen hast
3. Überlege, ob ein vergessener Baustein das Ergebnis verbessert hätte

Du wirst merken: Die meisten Leute vergessen konsistent dieselben Bausteine. Das ist genau der Punkt, an dem Frameworks helfen.

Kapitel 2: Zero-Shot Prompting – Ohne Beispiel zum Ziel

Zero-Shot Prompting ist das, was du bisher die ganze Zeit gemacht hast. Du gibst dem Modell eine Aufgabe – ohne Beispiel, ohne Vorlage, ohne “So soll das Ergebnis aussehen”. Einfach: Aufgabe rein, Ergebnis raus.

Klingt unspektakulär? Ist es auch. Aber es lohnt sich, bewusst darüber nachzudenken, weil Zero-Shot die Basis für alles Weitere ist.

Was Zero-Shot bedeutet

“Shot” kommt aus dem Englischen und bedeutet hier so viel wie “Beispiel”. “Zero Shots” heißt: null Beispiele.

Du sagst dem Modell, *was* es tun soll. Aber du zeigst ihm nicht, *wie* das Ergebnis aussehen soll.

Beispiel für Zero-Shot:

Fasse den folgenden Text in drei Sätzen zusammen:

[Text hier einfügen]

Du gibst keine Beispiel-Zusammenfassung. Du vertraust darauf, dass das Modell weiß, was “zusammenfassen” bedeutet und wie eine gute Zusammenfassung aussieht.

Und meistens klappt das. Erstaunlich gut sogar.

Warum Zero-Shot oft reicht

Moderne LLMs wie GPT-4, Claude 4 oder Gemini sind so gut vortrainiert, dass sie die meisten Standardaufgaben ohne Beispiele bewältigen. Sie haben während ihres Trainings Millionen von Zusammenfassungen, Übersetzungen, Analysen und Texten gesehen. Sie “wissen”, wie diese Formate aussehen.

Zero-Shot ist ideal, wenn:

- Die Aufgabe klar und eindeutig ist
- Du ein Standardformat erwartest (Liste, Zusammenfassung, Übersetzung)
- Du schnell ein Ergebnis brauchst und keine Zeit für aufwendige Prompts hast
- Das Thema zum Allgemeinwissen gehört

Gute Zero-Shot-Aufgaben:

- “Übersetze diesen Text ins Englische”
- “Schreibe eine Betreffzeile für eine E-Mail über Projektverzögerung”
- “Erkläre Photosynthese für einen 10-Jährigen”
- “Liste 5 Vorteile von Remote-Arbeit auf”

Wann Zero-Shot nicht reicht

Es gibt Situationen, in denen Zero-Shot an seine Grenzen stößt:

1. Ungewöhnliche Formate

Wenn du ein sehr spezifisches Format brauchst, das nicht zum Standard gehört, wird Zero-Shot unpräzise.

```
Schreib eine Produktbeschreibung im Stil einer Weinverkostungs-Notiz für ein Softwareprodukt.
```

Das Modell wird etwas produzieren, aber ob es deinem Geschmack entspricht? Unwahrscheinlich. Hier wäre ein Beispiel (One-Shot) hilfreicher.

2. Konsistente Ausgaben

Wenn du mehrere Texte brauchst, die alle gleich aufgebaut sein sollen, ist Zero-Shot riskant. Jedes Mal entscheidet das Modell neu, wie es die Aufgabe angeht. Mal ist die Zusammenfassung 2 Sätze lang, mal 5. Mal benutzt es Aufzählungszeichen, mal nicht.

3. Fachspezifische Aufgaben

Bei sehr spezialisierten Aufgaben – juristisches Schreiben, medizinische Dokumentation, technische Spezifikationen – fehlt dem Modell manchmal der genaue Kontext, wie das Ergebnis in *deinem* Fachgebiet aussehen soll.

4. Subjektive Qualität

Wenn “gut” eine Frage des persönlichen Geschmacks ist, kann das Modell nicht erraten, was du meinst. Soll der Blogartikel witzig sein oder seriös? Kurze Sätze oder verschachtelte? Ohne Beispiel ist das Glückssache.

Zero-Shot mit guter Struktur

Nur weil du keine Beispiele gibst, heißt das nicht, dass dein Prompt kurz sein muss. Du kannst Zero-Shot-Prompts trotzdem strukturiert aufbauen. Tatsächlich sind die besten Zero-Shot-Prompts sehr detailliert – nur eben ohne Beispiel-Output.

Schwacher Zero-Shot:

Schreib eine E-Mail an einen Kunden.

Starker Zero-Shot:

Schreibe eine professionelle E-Mail an einen Bestandskunden, der sich über eine verspätete Lieferung beschwert hat.

Inhalt: Entschuldigung, Erklärung (Lieferketten-Problem), konkrete Lösung (Expresslieferung auf unsere Kosten + 10% Gutschein).

Ton: Verständnisvoll, lösungsorientiert, nicht unterwürfig.

Länge: 150-200 Wörter.

Absender: Kundenservice-Team der Firma TechShop.

Das ist immer noch Zero-Shot – kein Beispiel einer fertigen E-Mail. Aber der Prompt gibt dem Modell genug Information, um ein brauchbares Ergebnis zu produzieren.

Die Zero-Shot-Faustregel

Je klarer deine Aufgabenbeschreibung, desto weniger brauchst du Beispiele. Umgekehrt: Je vager dein Prompt, desto mehr brauchst du Beispiele, um die Lücke zu füllen.

Das lässt sich als Formel ausdrücken:

Prompt-Qualität = Klarheit der Anweisung + Qualität der Beispiele

Bei Zero-Shot muss die Klarheit der Anweisung die ganze Arbeit machen. Und das geht – wenn du die Grundlagen aus Band 1 anwendest.

Zero-Shot in der Praxis: 5 Templates

Hier sind fünf Zero-Shot-Templates, die du sofort verwenden kannst:

Template 1: Zusammenfassung

Fasse den folgenden Text zusammen.
Maximal [X] Sätze.
Fokus auf: [Hauptthema/Aspekt].
Zielgruppe: [Wer soll das lesen?].

Text:
[Text einfügen]

Template 2: Analyse

Analysiere den folgenden [Text/Datensatz/Sachverhalt].
Fokussiere dich auf: [Aspekte].
Strukturiere deine Analyse in:
1. Zusammenfassung (2-3 Sätze)
2. Hauptkenntnisse (Aufzählung)
3. Empfehlung (1-2 Sätze)

[Inhalt einfügen]

Template 3: Umschreiben

Schreibe den folgenden Text um.
Zielgruppe: [Wer?]
Ton: [Welcher Stil?]
Länge: [Ungefähr wie lang?]
Behalte bei: [Was soll gleich bleiben?]

Originaltext:
[Text einfügen]

Template 4: Brainstorming

Generiere [X] Ideen für [Thema].
Kontext: [Wofür brauchst du die Ideen?]
Einschränkungen: [Budget, Zeitrahmen, Ressourcen]
Format: Nummerierte Liste mit jeweils 1-2 Sätzen Erklärung.

Template 5: Erklärung

Erkläre [Konzept] so, dass [Zielgruppe] es versteht.
Verwende [Analogien/Beispiele/keine Fachbegriffe].
Länge: [X] Wörter.
Beginne mit dem wichtigsten Punkt.

Zusammenfassung

- Zero-Shot = Aufgabe ohne Beispiel
- Funktioniert gut bei Standardaufgaben und klaren Anweisungen
- Schwächen bei ungewöhnlichen Formaten und subjektiver Qualität
- Je klarer dein Prompt, desto besser das Ergebnis ohne Beispiele

- Die meisten deiner Alltagsprompts werden Zero-Shot sein – und das ist völlig okay
-

Übung

Zero-Shot Challenge

Schreibe drei Zero-Shot-Prompts für folgende Aufgaben. Nutze dabei mindestens 3 der 5 Bausteine aus Band 1 (Aufgabe, Kontext, Format, Ton, Einschränkungen):

1. Eine LinkedIn-Post-Idee für dein Fachgebiet
2. Eine Erklärung von Blockchain für deine Großmutter
3. Drei Verbesserungsvorschläge für eine Teammeeting-Kultur

Teste die Prompts in einem LLM deiner Wahl. Bewerte die Ergebnisse auf einer Skala von 1-10. Merke dir die Bewertung – in den nächsten Kapiteln vergleichst du sie mit One-Shot und Few-Shot Varianten derselben Aufgaben.

Kapitel 3: One-Shot Prompting

– Ein Beispiel sagt mehr als tausend Worte

Du kennst das Sprichwort. Und bei Prompting stimmt es ganz besonders.

One-Shot Prompting bedeutet: Du gibst dem Modell genau ein Beispiel, bevor du die eigentliche Aufgabe stellst. Ein einziges Beispiel, das zeigt: “So soll das Ergebnis aussehen.”

Warum ein einziges Beispiel so viel bewirkt

Stell dir vor, jemand bittet dich: “Schreib eine Produktbewertung.” Du würdest wahrscheinlich fragen: “Wie soll die aussehen? Lang? Kurz? Mit Sternen? Formal?”

Aber wenn jemand sagt: “Schreib eine Produktbewertung. Hier ist ein Beispiel: *‘Der Akku hält ewig, die Kamera ist solide, aber die App stürzt regelmäßig ab. 3,5/5 Sterne.’* Jetzt schreib eine für Kopfhörer.” – dann weißt du sofort, was gemeint ist. Länge, Stil, Struktur – alles klar.

Genau so funktioniert One-Shot bei LLMs. Das Modell erkennt das Muster im Beispiel und repliziert es.

Die Anatomie eines One-Shot-Prompts

Ein One-Shot-Prompt hat drei Teile:

1. **Aufgabenbeschreibung** – Was soll das Modell tun?
2. **Beispiel** – Ein konkretes Input-Output-Paar
3. **Eigentliche Aufgabe** – Die Aufgabe, die das Modell lösen soll

Struktur:

[Aufgabenbeschreibung]

Beispiel:

Input: [Beispiel-Eingabe]

Output: [Beispiel-Ausgabe]

Aufgabe:

Input: [Deine tatsächliche Eingabe]

Output:

One-Shot in Aktion

Beispiel 1: Produktbeschreibungen

Schreibe kurze, ansprechende Produktbeschreibungen für einen Online-Shop.

Beispiel:

Produkt: Thermobecher 500ml, Edelstahl, doppelwandig

Beschreibung: Dein Kaffee bleibt heiß, deine Hand bleibt kühl.

Der doppelwandige Edelstahl-Thermobecher hält Getränke bis zu

6 Stunden warm - perfekt für lange Meetings und Pendler-Morgen.

BPA-frei und spülmaschinengeeignet.

Aufgabe:

Produkt: Bluetooth-Kopfhörer, Over-Ear, Active Noise Cancelling, 30h Akku

Beschreibung:

Das Modell versteht sofort: Es soll eine ähnlich lange, ähnlich aufgebaute Beschreibung schreiben. Mit Benefit-Sprache. Mit konkreten Details.

Beispiel 2: Daten extrahieren

Extrahiere strukturierte Daten aus Kundenbewertungen.

Beispiel:

Bewertung: "Super schnelle Lieferung, aber die Verpackung war beschädigt. Das Produkt selbst funktioniert einwandfrei. Würde wieder bestellen."

Extraktion:

- Lieferung: positiv (schnell)
- Verpackung: negativ (beschädigt)
- Produkt: positiv (funktioniert)
- Wiederkauf: ja

Aufgabe:

Bewertung: "Habe den Artikel nach 2 Wochen zurückgeschickt. Die Farbe war komplett anders als auf dem Foto. Der Kundenservice war allerdings sehr hilfsbereit und hat sofort erstattet."

Extraktion:

Ohne das Beispiel hätte das Modell die Daten vielleicht als Fließtext ausgegeben, oder mit anderen Kategorien, oder in einem völlig anderen Format. Das Beispiel definiert das Schema.

Beispiel 3: Tonalität treffen

Schreibe Social-Media-Posts im Stil unserer Marke.

Beispiel:

Thema: Neues Feature – Dark Mode

Post: Dark Mode ist da. Endlich. Eure Augen können uns danken.

Ab sofort in der App. Update und ausprobieren. 🌙

Aufgabe:

Thema: Neues Feature – Offline-Modus

Post:

Hier zeigt das Beispiel nicht nur das Format, sondern den Ton. Kurze Sätze. Frech. Direkt. Das Modell übernimmt diesen Stil.

Wann One-Shot besser ist als Zero-Shot

One-Shot lohnt sich besonders in diesen Situationen:

Ungewöhnliche Formate

Wenn du ein Format brauchst, das nicht zum Standard gehört. Statt es umständlich zu beschreiben, zeigst du es einfach.

Konsistenz

Wenn du mehrere ähnliche Texte brauchst, die alle gleich aufgebaut sein sollen. Das Beispiel definiert den Standard.

Subjektive Qualität

Wenn “gut” eine Geschmacksfrage ist. Statt zu beschreiben, wie der Text klingen soll, zeigst du es.

Komplexe Transformationen

Wenn du Daten von einem Format in ein anderes umwandeln willst. Ein Beispiel ist klarer als jede Beschreibung.

Die Qualität des Beispiels entscheidet

Das ist der wichtigste Punkt in diesem Kapitel: Dein Beispiel muss gut sein. Wenn du ein mittelmäßiges Beispiel gibst, bekommst du mittelmäßige Ergebnisse. Das Modell repliziert, was es sieht – inklusive der Schwächen.

Kriterien für ein gutes Beispiel:

1. **Repräsentativ** – Es sollte typisch für die Aufgabe sein, nicht ein Sonderfall
2. **Hochwertig** – Es sollte das Qualitätsniveau zeigen, das du erwartest
3. **Vollständig** – Es sollte alle Elemente enthalten, die du im Output willst
4. **Eindeutig** – Es sollte kein Rauschen oder irrelevante Elemente enthalten

Schlechtes Beispiel:

```
Beispiel:  
Eingabe: Hund  
Ausgabe: Ein Hund ist ein Tier das bellt und als haustier  
gehalten wird es gibt viele Rassen zum beispiel Pudel oder  
Schäferhund
```

Gutes Beispiel:

Beispiel:
Eingabe: Hund
Ausgabe: Ein Hund (*Canis lupus familiaris*) ist ein domestiziertes Säugetier aus der Familie der Canidae. Als eines der ältesten Haustiere des Menschen gibt es heute über 350 anerkannte Rassen, von kleinen Chihuahuas bis zu großen Deutschen Doggen.

Du siehst den Unterschied. Das erste Beispiel hat keine Satzzeichen, keine Struktur und keine Tiefe. Wenn du das als Vorlage gibst, wundere dich nicht über schlampige Ergebnisse.

One-Shot vs. Längere Anweisungen

Manchmal fragen sich Leute: “Soll ich lieber ein Beispiel geben oder lieber genauer beschreiben, was ich will?”

Die Antwort: Es kommt drauf an.

Situation	Besserer Ansatz
Format ist schwer zu beschreiben	One-Shot (zeigen statt erklären)
Stil/Ton soll getroffen werden	One-Shot (Beispiel definiert den Ton)
Logik/Regeln sollen befolgt werden	Anweisung (klare Regeln formulieren)
Beides nötig	Kombination (Regeln + Beispiel)

Und ja, du kannst beides kombinieren. Ein Prompt mit klaren Anweisungen UND einem Beispiel ist oft die beste Lösung.

Zusammenfassung

- One-Shot = Eine Aufgabe mit genau einem Beispiel
 - Das Beispiel definiert Format, Stil, Struktur und Qualitätsniveau
 - Funktioniert besser als Zero-Shot bei ungewöhnlichen Formaten und subjektiver Qualität
 - Die Qualität des Beispiels bestimmt die Qualität des Ergebnisses
 - Kann mit Anweisungen kombiniert werden
-

Übung

One-Shot Upgrade

Nimm die drei Zero-Shot-Prompts aus der Übung des letzten Kapitels und mache daraus One-Shot-Prompts:

1. Schreibe zuerst selbst ein Beispiel für das gewünschte Ergebnis
2. Füge es in den Prompt ein
3. Teste den neuen Prompt im selben LLM
4. Bewerte das Ergebnis auf der gleichen 1-10 Skala
5. Vergleiche: Hat sich die Qualität verbessert?

Besonders spannend: Teste den gleichen One-Shot-Prompt mit verschiedenen LLMs (ChatGPT, Claude, Gemini). Wo macht das Beispiel den größten Unterschied?

Kapitel 4: Few-Shot Prompting

– Muster erkennen durch mehrere Beispiele

Wenn ein Beispiel gut ist, sind mehrere besser. Zumindest meistens.

Few-Shot Prompting ist die natürliche Weiterentwicklung von One-Shot. Statt einem Beispiel gibst du dem Modell zwei, drei oder mehr. Das Modell erkennt das Muster zwischen den Beispielen und wendet es auf die neue Aufgabe an.

Wie Few-Shot funktioniert

Das Prinzip ist simpel: Je mehr Beispiele du gibst, desto besser versteht das Modell, was du willst. Nicht weil es “klüger” wird, sondern weil es mehr Datenpunkte hat, aus denen es Muster ableiten kann.

Stell dir vor, ich sage dir: “Übersetze ‘Haus’ ins Klingonische.” Du hast keine Ahnung. Aber wenn ich dir sage:

- Wasser → biQ
- Feuer → qul
- Erde → tera’

Dann hast du zumindest ein Gefühl für die Sprache. Du kennst die Schreibweise, die Länge, den Stil. Du könntest raten. Und LLMs sind verdammt gut im Raten.

Die Struktur eines Few-Shot-Prompts

[Aufgabenbeschreibung]

Beispiel 1:

Input: [Eingabe 1]

Output: [Ausgabe 1]

Beispiel 2:

Input: [Eingabe 2]

Output: [Ausgabe 2]

Beispiel 3:

Input: [Eingabe 3]

Output: [Ausgabe 3]

Aufgabe:

Input: [Deine Eingabe]

Output:

Few-Shot in Aktion

Beispiel 1: Sentiment-Analyse

Klassifiziere Kundenbewertungen als positiv, neutral oder negativ.

Beispiel 1:

Bewertung: "Absolut klasse! Schnelle Lieferung und top Qualität."

Sentiment: positiv

Beispiel 2:

Bewertung: "Ist okay, nichts Besonderes. Erfüllt seinen Zweck."

Sentiment: neutral

Beispiel 3:

Bewertung: "Totaler Reinfl. Nach einer Woche kaputt gegangen."

Sentiment: negativ

Beispiel 4:

Bewertung: "Die Lieferung hat ewig gedauert, aber das Produkt selbst ist wirklich gut."

Sentiment: neutral

Aufgabe:

Bewertung: "Hatte Bedenken wegen der vielen negativen Bewertungen, aber bei mir funktioniert alles einwandfrei. Preis-Leistung top."

Sentiment:

Beachte Beispiel 4 – eine gemischte Bewertung. Durch dieses Beispiel zeigt du dem Modell, wie es mit Ambiguität umgehen soll. Ohne dieses Beispiel hätte es vielleicht "positiv" gesagt, weil das Produkt gelobt wird. Oder "negativ", weil die Lieferung kritisiert wird. Das Beispiel kalibriert die Erwartung.

Beispiel 2: Formatierung

Wandle informelle Notizen in strukturierte Meeting-Protokoll-Einträge um.

Beispiel 1:

Notiz: "müssen das budget für q3 nochmal besprechen, sarah kümmert sich um die zahlen bis freitag"

Protokoll:

```
**Thema:** Q3-Budget-Review
**Beschluss:** Erneute Besprechung erforderlich
**Verantwortlich:** Sarah
**Deadline:** Freitag
**Status:** Offen
```

Beispiel 2:

Notiz: "website relaunch wird auf januar verschoben wegen personalengpass, tom informiert den kunden"

Protokoll:

```
**Thema:** Website-Relaunch Terminverschiebung
**Beschluss:** Verschiebung auf Januar (Grund: Personalengpass)
**Verantwortlich:** Tom (Kundenkommunikation)
**Deadline:** Januar
**Status:** Verschoben
```

Aufgabe:

Notiz: "neue social media strategie steht, lisa und markus setzen

die erste kampagne bis ende des monats um, budget ist genehmigt"

Protokoll:

Hier arbeiten die Beispiele zusammen: Beispiel 1 zeigt eine offene Aufgabe, Beispiel 2 zeigt eine Verschiebung. Zusammen definieren sie das Schema vollständig.

Beispiel 3: Kreative Konsistenz

Schreibe Kurzbeschreibungen für Podcast-Episoden in unserem Stil.

Beispiel 1:

Titel: "Warum Meetings meistens Zeitverschwendung sind"

Beschreibung: 47 Minuten über die Meeting-Kultur in deutschen

Unternehmen. Warum die meisten Meetings E-Mails sein sollten, welche Meetings tatsächlich Sinn machen, und wie du deinen Chef

überzeugst, dass "kurz synchronisieren" kein Synonym für "eine Stunde quatschen" ist.

Beispiel 2:

Titel: "Remote Work – zwei Jahre später"

Beschreibung: 52 Minuten ehrliche Bilanz. Was funktioniert hat,

was nicht, und warum die "Zurück-ins-Büro"-Debatte an der Realität vorbeigeht. Mit echten Zahlen statt Bauchgefühl.

Aufgabe:

Titel: "KI im Arbeitsalltag – zwischen Hype und Realität"

Beschreibung:

Durch zwei Beispiele versteht das Modell: Dauer erwähnen, provokante Tonlage, konkret statt vage, kritisch aber nicht negativ.

Wie viele Beispiele sind genug?

Die kurze Antwort: Zwischen 2 und 5.

Die längere Antwort:

Anzahl	Wann sinnvoll
2 Beispiele	Einfache Aufgaben, klares Muster
3 Beispiele	Standardfall, reicht für die meisten Aufgaben
4-5 Beispiele	Komplexe Aufgaben, Grenzfälle zeigen
6+ Beispiele	Selten nötig, kann sogar kontraproduktiv sein

Warum nicht mehr? Zwei Gründe:

1. **Kontextfenster:** Jedes Beispiel verbraucht Token. Bei 10 langen Beispielen ist der halbe Kontext aufgebraucht, bevor die eigentliche Aufgabe kommt.
2. **Abnehmender Grenznutzen:** Der Sprung von 0 auf 1 Beispiel ist riesig. Von 1 auf 3 ist merkbar. Von 3 auf 5 ist minimal. Von 5 auf 10 ist oft Null.

Die Kunst der Beispielauswahl

Nicht alle Beispiele sind gleich wertvoll. Die besten Few-Shot-Sets folgen diesen Regeln:

Regel 1: Vielfalt

Deine Beispiele sollten verschiedene Varianten der Aufgabe zeigen, nicht dreimal dasselbe.

Schlecht: Drei Beispiele, die alle positives Sentiment zeigen.

Gut: Ein positives, ein negatives, ein gemischtes Beispiel.

Regel 2: Grenzfälle einschließen

Zeige nicht nur die einfachen Fälle. Zeige auch die schwierigen.

Schlecht: Nur eindeutige Beispiele.

Gut: Auch ein Beispiel, das ambig ist, um zu zeigen, wie das Modell damit umgehen soll.

Regel 3: Reihenfolge beachten

Forschungen zeigen, dass die Reihenfolge der Beispiele das Ergebnis beeinflusst. Als Faustregel: Setze das Beispiel, das der aktuellen Aufgabe am ähnlichsten ist, an die letzte Position – direkt vor deine eigentliche Aufgabe.

Regel 4: Konsistente Formatierung

Alle Beispiele müssen gleich formatiert sein. Wenn Beispiel 1 Bullet Points hat und Beispiel 2 nummerierte Listen, verwirrt das das Modell.

Few-Shot vs. Fine-Tuning

Kurzer Ausblick für Neugierige: Few-Shot Prompting ist eine Art “Mini-Training” im Prompt selbst. Du gibst dem Modell Beispiele, es lernt daraus – aber nur für diese eine Anfrage. Beim nächsten Prompt ist alles vergessen.

Fine-Tuning dagegen verändert das Modell dauerhaft. Du trainierst es mit hunderten oder tausenden Beispielen, und es behält das Gelernte. Das ist mächtiger, aber auch aufwendiger und teurer.

Für die meisten Anwendungen reicht Few-Shot völlig aus. Fine-Tuning ist etwas für Band 7, wenn wir über Entwickler-Themen sprechen.

Zusammenfassung

- Few-Shot = Aufgabe mit mehreren Beispielen (typischerweise 2-5)
- Ideal für Klassifikation, Formatierung und konsistente Outputs
- Die Qualität und Vielfalt der Beispiele ist entscheidend

- 3 gute Beispiele schlagen 10 mittelmäßige
 - Grenzfälle in den Beispielen verbessern die Ergebnisse deutlich
-

Übung

Few-Shot Meisterklasse

Erstelle einen Few-Shot-Prompt für folgende Aufgabe: Du willst, dass die KI aus unstrukturierten Tagebuch-Einträgen strukturierte Stimmungs-Logs erstellt.

1. Schreibe mindestens 3 Beispiel-Paare (Tagebuch-Eintrag → Stimmungs-Log)
2. Einer der Einträge sollte gemischte Stimmungen enthalten
3. Definiere das Format des Stimmungs-Logs selbst
4. Teste den Prompt mit einem echten (oder ausgedachten) Tagebuch-Eintrag
5. Vergleiche: Wie wäre das Ergebnis ohne Beispiele ausgefallen?

Bonus: Teste denselben Prompt einmal mit 2 Beispielen und einmal mit 4. Merkst du einen Unterschied?

Kapitel 5: Das CRAFT-Framework – Dein Allrounder

Jetzt wird's ernst. Wir verlassen die Shot-Typen und steigen in die Struktur-Frameworks ein. Und wir fangen mit dem vielseitigsten an: CRAFT.

CRAFT steht für:

- **C** – Context (Kontext)
- **R** – Role (Rolle)
- **A** – Action (Aufgabe/Aktion)
- **F** – Format (Ausgabeformat)
- **T** – Tone (Tonalität)

Kommt dir bekannt vor? Sollte es. Das sind im Grunde die 5 Bausteine aus Band 1, nur anders sortiert und unter einem einprägsamen Namen verpackt.

Warum CRAFT?

Weil es funktioniert. Und weil es leicht zu merken ist.

Ich habe dutzende Frameworks getestet. Manche hatten 8 Buchstaben, manche 3. CRAFT ist der Sweet Spot. Es ist detailliert genug, um fast jede Aufgabe abzudecken, und kompakt genug, um es sich tatsächlich zu merken.

Außerdem passt der Name: CRAFT bedeutet auf Englisch "Handwerk". Und genau das ist Prompting – ein Handwerk.

Die fünf Elemente im Detail

C – Context (Kontext)

Der Kontext beantwortet die Frage: Was ist die Ausgangssituation?

Hier gibst du dem Modell alle Hintergrundinformationen, die es braucht, um deine Aufgabe zu verstehen. Wer bist du? Wofür brauchst du das Ergebnis? Welche Rahmenbedingungen gibt es?

Beispiele:

- “Ich bin Teamleiterin in einem mittelständischen IT-Unternehmen mit 50 Mitarbeitern.”
- “Wir planen ein Webinar zum Thema Datenschutz für kleine Unternehmen.”
- “Der Kunde hat sich dreimal beschwert und droht mit Kündigung.”

R – Role (Rolle)

Die Rolle beantwortet die Frage: Wer soll das Modell sein?

In Band 1 hast du gelernt, dass Rollen die Qualität der Antwort massiv beeinflussen. In CRAFT ist die Rolle fest im Framework verankert – du vergisst sie nicht mehr.

Beispiele:

- “Du bist ein erfahrener HR-Manager mit 15 Jahren Erfahrung.”
- “Du bist eine Marketingexpertin, die sich auf B2B-SaaS spezialisiert hat.”
- “Du bist ein geduldiger Nachhilflehrer für Mathematik.”

A – Action (Aktion)

Die Aktion beantwortet die Frage: Was genau soll das Modell tun?

Das ist der Kern deines Prompts. Klare, eindeutige Anweisung. Kein “Hilf mir mal” oder “Mach was Schönes”. Sondern: “Schreibe”, “Analysiere”, “Erstelle”, “Vergleiche”.

Beispiele:

- “Schreibe eine Einladungs-E-Mail für das Webinar.”
- “Erstelle eine Pro-Contra-Liste für die Einführung von Remote Work.”
- “Analysiere diese drei Bewerbungen und erstelle eine Ranking-Liste.”

F – Format (Format)

Das Format beantwortet die Frage: Wie soll das Ergebnis aussehen?

Ohne Format-Vorgabe entscheidet das Modell selbst – und die Entscheidung ist selten optimal. Mit Format-Vorgabe bekommst du genau das, was du brauchst.

Beispiele:

- “Bullet-Point-Liste mit maximal 7 Punkten”
- “Tabelle mit Spalten: Name, Stärken, Schwächen, Gesamtbewertung”
- “Fließtext, 300-400 Wörter, mit Zwischenüberschriften”
- “JSON-Objekt mit den Feldern: Titel, Zusammenfassung, Keywords”

T – Tone (Tonalität)

Der Ton beantwortet die Frage: Wie soll es klingen?

Der am meisten unterschätzte Baustein. Viele Leute denken, der Ton sei egal, weil es “nur eine KI” ist. Aber der Ton bestimmt, ob dein Ergebnis nach Pressemitteilung klingt oder nach Gespräch unter Freunden.

Beispiele:

- “Professionell aber zugänglich, keine Fachsprache”
- “Freundlich und ermutigend, wie ein guter Coach”
- “Direkt und sachlich, ohne Floskeln”
- “Humorvoll und leicht, wie ein Podcast-Gespräch”

CRAFT in der Praxis: Vollständige Beispiele

Beispiel 1: Blog-Artikel

Context: Ich betreibe einen Karriere-Blog für Young Professionals (25-35 Jahre) in der DACH-Region. Der Blog hat monatlich ca. 15.000 Leser. Thema dieses Monats: Gehaltsverhandlung.

Role: Du bist ein erfahrener Karrierecoach, der seit 10 Jahren Young Professionals bei Gehaltsverhandlungen berät. Du kennst die typischen Fehler und hast ein Händchen für praxisnahe Tipps.

Action: Schreibe einen Blogartikel über die 5 größten Fehler bei der ersten Gehaltsverhandlung und wie man sie vermeidet.

Format: 1200-1500 Wörter. Einleitung (Hook + Problemstellung), 5 Fehler als nummerierte Abschnitte (je mit Beispielsituation und konkretem Tipp), Fazit mit Ermutigung. Verwende Zwischenüberschriften (H2 für jeden Fehler).

Tone: Locker aber kompetent. Du-Ansprache. Konkrete Beispiele statt abstrakte Ratschläge. Kein "Sei einfach selbstbewusst"-Gelaber.

Beispiel 2: Kundenservice-Antwort

Context: Ich arbeite im Kundenservice eines Online-Möbelhauses.

Ein Kunde hat einen Schreibtisch bestellt, der mit einer verkratzten Oberfläche ankam. Es ist sein zweiter Reklamationsfall in 3 Monaten.

Role: Du bist ein erfahrener Kundenservice-Mitarbeiter, der Eskalationen mit Empathie und konkreten Lösungen entschärft.

Action: Verfasse eine Antwort-E-Mail, die das Problem anerkennt, sich entschuldigt und eine Lösung anbietet (Austauschlieferrung + 15% Gutschein auf die nächste Bestellung als Wiedergutmachung).

Format: Maximal 200 Wörter. Anrede, 3-4 kurze Absätze, Grußformel.

Keine Standard-Textbausteine wie "Ihr Anliegen ist uns wichtig".

Tone: Aufrichtig entschuldigend, ohne übertrieben unterwürfig zu sein.

Lösungsorientiert. Der Kunde soll das Gefühl haben, gehört zu werden.

Beispiel 3: Technische Dokumentation

Context: Wir haben gerade eine neue REST-API für unser Projektmanagement-Tool gelauncht. Die Dokumentation soll Entwicklern helfen, die API schnell zu integrieren.

Role: Du bist ein Technical Writer mit Erfahrung in API-Dokumentation.

Du schreibst für Entwickler, die wenig Zeit haben und schnell funktionierende Beispiele brauchen.

Action: Schreibe die Dokumentation für den Endpunkt POST /api/v1/projects. Decke ab: Beschreibung, Authentifizierung, Request-Parameter, Beispiel-Request, Beispiel-Response, Fehlercodes.

Format: Markdown. Klare Überschriften, Code-Blöcke für Beispiele (curl + Python), Tabelle für Parameter und Fehlercodes.

Tone: Sachlich und präzise. Kein Marketing-Sprech. Direkt und technisch korrekt, aber nicht trocken.

CRAFT muss nicht immer vollständig sein

Wichtiger Punkt: Du musst nicht immer alle fünf Elemente verwenden. CRAFT ist eine Checkliste, kein Formular, das komplett ausgefüllt werden muss.

- Schnelle Frage an die KI? Vielleicht reicht A + F.
- Kreativer Text? C + R + A + T ist wichtiger als F.
- Datenaufgabe? C + A + F, ohne R und T.

Die Faustregel: Je komplexer die Aufgabe, desto mehr CRAFT-Elemente brauchst du.

Komplexität	Empfohlene Elemente
Einfach (Übersetzung, Zusammenfassung)	A, F
Mittel (E-Mail, Blogartikel)	C, R, A, F, T
Komplex (Strategie, Analyse)	Alle + Einschränkungen

CRAFT kombiniert mit Shot-Typen

Und hier wird's richtig mächtig: Du kannst CRAFT mit Zero-Shot, One-Shot oder Few-Shot kombinieren.

CRAFT + Zero-Shot: Der Prompt oben. Klare Struktur, keine Beispiele.

CRAFT + One-Shot: Du fügst nach der CRAFT-Beschreibung ein Beispiel hinzu.

[CRAFT-Prompt wie oben]

Hier ist ein Beispiel für den gewünschten Stil:
[Beispiel einfügen]

Jetzt schreibe den Text für das neue Thema:
[Thema]

CRAFT + Few-Shot: Mehrere Beispiele nach dem CRAFT-Block.

Das ist die Profi-Variante. Und ja, die Prompts werden dabei lang. Das ist okay. Ein langer, strukturierter Prompt schlägt einen kurzen, vagen Prompt jedes Mal.

Zusammenfassung

- CRAFT = Context, Role, Action, Format, Tone

- Das vielseitigste Framework – funktioniert für fast jede Aufgabe
 - Nicht immer alle 5 Elemente nötig – anpassen je nach Komplexität
 - Kombinierbar mit Shot-Typen für maximale Kontrolle
 - CRAFT ist der logische nächste Schritt nach den 5 Bausteinen aus Band 1
-

Übung

CRAFT-Marathon

Schreibe drei vollständige CRAFT-Prompts für folgende Szenarien:

1. **Bewerbung:** Du willst ein Motivationsschreiben für deinen Traumjob erstellen lassen
2. **Präsentation:** Du brauchst eine Gliederung für einen Vortrag vor dem Management
3. **Social Media:** Du willst eine Woche Instagram-Posts für ein kleines Café planen

Für jeden Prompt: Fülle alle 5 CRAFT-Elemente aus. Teste sie. Bewerte die Ergebnisse. Welches Element hat den größten Einfluss auf die Qualität?

Kapitel 6: Das RTF-Framework – Schnell und effektiv

Nicht jeder Prompt braucht ein fünfteiliges Framework. Manchmal willst du einfach schnell ein gutes Ergebnis, ohne jedes Mal einen halben Aufsatz zu schreiben.

Dafür gibt es RTF.

Was ist RTF?

RTF steht für:

- **R** – Role (Rolle)
- **T** – Task (Aufgabe)
- **F** – Format (Format)

Drei Elemente. Mehr nicht. RTF ist das Taschenmesser unter den Frameworks – kompakt, praktisch, immer dabei.

Warum nur drei Elemente?

Weil drei oft reichen.

Denk mal drüber nach: Wenn du einem Kollegen eine Aufgabe gibst, sagst du meistens drei Dinge: Wer soll was machen und wie soll das Ergebnis aussehen. Du sagst nicht jedes Mal den kompletten Kontext deines Unternehmens und die gewünschte emotionale Tonalität dazu.

RTF basiert auf derselben Logik. Es deckt die drei wichtigsten Entscheidungen ab:

1. Aus welcher Perspektive soll geantwortet werden? (Role)
2. Was genau soll getan werden? (Task)
3. Wie soll das Ergebnis aussehen? (Format)

Kontext und Ton? Kannst du in die Task-Beschreibung einbauen, wenn nötig. Oder weglassen, wenn es auch ohne geht.

RTF in Aktion

Beispiel 1: Schnelle E-Mail

Role: Professioneller Business-Kommunikator

Task: Schreibe eine kurze E-Mail an das Team, die über die Verschiebung des Projekt-Deadlines von Freitag auf nächsten Mittwoch informiert. Grund: Feedback vom Kunden kam verspätet.

Format: 3-4 Sätze, Betreff + E-Mail-Text

Kein langer Kontext über die Firma. Kein ausführlicher Ton-Guide. Reicht trotzdem für eine brauchbare E-Mail.

Beispiel 2: Code-Hilfe

Role: Senior Python-Entwickler
Task: Schreibe eine Funktion, die eine CSV-Datei einliest und alle Zeilen zurückgibt, in denen der Wert in Spalte "Status" gleich "aktiv" ist.
Format: Python-Code mit Docstring und Typ-Hints

Beispiel 3: Lernhilfe

Role: Geduldiger Mathe-Nachhilfelehrer für Oberstufenschüler
Task: Erkläre den Satz des Pythagoras mit einem Alltagsbeispiel.
Format: Maximal 5 Sätze, dann ein konkretes Rechenbeispiel

Beispiel 4: Content-Ideen

Role: Social-Media-Strategin für kleine Unternehmen
Task: Generiere 10 Instagram-Post-Ideen für eine lokale Bäckerei im Frühling.
Format: Nummerierte Liste, jeweils: Post-Idee (1 Satz) + empfohlenes Format (Reel/Karussell/Einzelbild)

RTF vs. CRAFT: Wann was?

Die ehrliche Antwort: RTF ist CRAFT ohne Kontext und Ton. Das macht es schneller, aber auch weniger präzise.

Kriterium	RTF	CRAFT
Geschwindigkeit	Schneller zu schreiben	Braucht mehr Denkarbeit
Präzision	Gut für Standardaufgaben	Besser für komplexe Aufgaben
Reproduzierbarkeit	Mittel	Hoch
Lernkurve	Niedrig	Mittel
Ideal für	Quick Tasks, tägliche Aufgaben	Wichtige Texte, wiederkehrende Workflows

Nutze RTF, wenn:

- Du schnell ein Ergebnis brauchst
- Die Aufgabe eindeutig ist
- Der Kontext offensichtlich ist
- Du den Output sowieso noch bearbeitest

Nutze CRAFT, wenn:

- Das Ergebnis direkt verwendbar sein soll
- Der Kontext komplex ist
- Der Ton entscheidend ist (Kundenansreiben, Marketing)
- Du den Prompt wiederverwenden willst

RTF als Einstieg

RTF ist auch ein guter Einstieg in die Welt der Frameworks. Wenn dir CRAFT zu viel ist – fang mit RTF an. Du wirst schnell merken, wann drei Elemente reichen und wann du mehr brauchst. Und dann erweiterst du schrittweise.

Ich benutze RTF für geschätzt 60% meiner täglichen Prompts. Die sind nicht für die Ewigkeit. Sie müssen funktionieren, nicht perfekt sein. Für die anderen 40% – die wichtigen, die wiederverwendbaren, die, bei denen die Qualität stimmen muss – nehme ich CRAFT oder RISEN.

RTF mit Erweiterungen

Du kannst RTF auch schrittweise erweitern, wenn du merkst, dass etwas fehlt:

RTF + Kontext:

Kontext: Startup mit 12 Mitarbeitern, informelle Kultur
Role: HR-Managerin
Task: Schreibe eine Stellenausschreibung für einen Front-end-Entwickler
Format: Maximal 300 Wörter, mit Bullet Points für Anforderungen

RTF + Ton:

Role: Tech-Journalist
Task: Schreibe eine Kurzkritik des neuen iPhone
Format: 150 Wörter, ein Absatz
Ton: Ehrlich, leicht sarkastisch, kein Fanboy-Sprech

RTF + Einschränkungen:

Role: Ernährungsberaterin
Task: Erstelle einen 7-Tage-Essensplan für eine vierköpfige Familie
Format: Tabelle (Tag | Frühstück | Mittagessen | Abendessen)
Einschränkungen: Vegetarisch, Budget max. 80€ pro Woche, keine Nüsse

Siehst du? RTF ist flexibel. Du startest mit drei Elementen und baust bei Bedarf aus. Das ist kein Verstoß gegen das Framework – das ist die richtige Art, es zu benutzen.

Zusammenfassung

- RTF = Role, Task, Format – das Minimal-Framework
 - Ideal für schnelle, alltägliche Prompts
 - Nicht für alles geeignet – bei komplexen Aufgaben fehlen Kontext und Ton
 - Flexibel erweiterbar bei Bedarf
 - Perfekt als Einstieg in Framework-basiertes Prompting
-

Übung

RTF-Sprint

Stelle eine Stoppuhr auf 10 Minuten. Schreibe in dieser Zeit so viele RTF-Prompts wie möglich für verschiedene Alltagsaufgaben:

- Eine E-Mail schreiben
- Ein Rezept finden
- Etwas erklären lassen
- Eine Entscheidung vorbereiten
- Einen Text übersetzen

Ziel: Mindestens 5 Prompts in 10 Minuten. Teste danach die zwei besten und bewerte: Reicht RTF, oder hättest du mehr Kontext gebraucht?

Kapitel 7: Das RISEN-Framework – Für die komplexen Fälle

Manchmal reicht “Schreib mir einen Text” nicht aus. Manchmal hast du eine Aufgabe, die mehrere Schritte hat, ein klares Ziel verfolgt und Grenzen braucht. Für genau solche Fälle gibt es RISEN.

Was ist RISEN?

RISEN steht für:

- **R** – Role (Rolle)
- **I** – Instructions (Anweisungen)
- **S** – Steps (Schritte)
- **E** – End Goal (Endziel)
- **N** – Narrowing (Eingrenzung)

Fünf Elemente, wie CRAFT. Aber mit einem anderen Fokus. Wo CRAFT den *Output* strukturiert (Format, Ton), strukturiert RISEN den *Prozess* (Schritte, Ziel, Grenzen).

Wann RISEN statt CRAFT?

RISEN ist dein Framework für Aufgaben, die:

- **Mehrstufig** sind – es gibt nicht nur ein “Mach das”, sondern eine Abfolge
- **Zielorientiert** – du weißt genau, wo du hinwillst
- **Begrenzt** werden müssen – das Modell soll nicht abschweifen

Typische RISEN-Aufgaben:

- Strategische Analysen
- Schritt-für-Schritt-Anleitungen
- Recherche mit klarem Ergebnis
- Komplexe Texte mit Struktur-Vorgaben
- Problemlösung mit definierten Rahmenbedingungen

Die fünf Elemente im Detail

R – Role (Rolle)

Wie bei CRAFT und RTF: Wer soll das Modell sein? Bei RISEN ist die Rolle oft spezifischer, weil die Aufgabe komplexer ist.

Role: Du bist ein erfahrener Unternehmensberater mit Spezialisierung auf digitale Transformation im Mittelstand. Du hast in den letzten 5 Jahren über 30 Projekte in der DACH-Region begleitet.

I – Instructions (Anweisungen)

Die Anweisungen sind der Kern: Was soll das Modell tun? Aber im Gegensatz zu CRAFTs “Action” sind die Instructions bei RISEN ausführlicher und können mehrere Teilaufgaben umfassen.

Instructions: Analysiere die folgende Geschäftssituation und entwickle eine Digitalstrategie. Berücksichtige dabei die aktuelle Marktposition, vorhandene Ressourcen und die Wettbewerbslandschaft. Identifiziere Quick Wins (umsetzbar in 3 Monaten) und langfristige Maßnahmen (6-12 Monate).

S – Steps (Schritte)

Hier wird RISEN einzigartig. Du gibst dem Modell eine explizite Schrittfolge vor. Das verhindert, dass das Modell Abkürzungen nimmt oder wichtige Aspekte überspringt.

- Steps:
1. Fasse die aktuelle Situation in 3-4 Sätzen zusammen
 2. Identifiziere die 3 größten Herausforderungen
 3. Entwickle für jede Herausforderung mindestens 2 Lösungsansätze
 4. Priorisiere die Ansätze nach Aufwand und Wirkung
 5. Erstelle einen konkreten Maßnahmenplan mit Zeitrahmen

E – End Goal (Endziel)

Das Endziel sagt dem Modell, wo es ankommen soll. Klingt redundant? Ist es nicht. Die Instructions sagen, *was* zu tun ist. Das End Goal sagt, *wozu*.

End Goal: Am Ende soll ein konkreter Maßnahmenplan stehen, den der Geschäftsführer in der nächsten Vorstandssitzung präsentieren kann. Der Plan muss ohne zusätzliche Erklärung verständlich und umsetzbar sein.

Warum das wichtig ist: Es zwingt das Modell, ergebnisorientiert zu arbeiten. Ohne End Goal produziert es vielleicht eine tolle Analyse, die aber nicht die Frage beantwortet, die der Geschäftsführer hat.

N – Narrowing (Eingrenzung)

Narrowing ist das, was ich in Band 1 “Einschränkungen” genannt habe – aber in RISEN hat es einen prominenteren Platz. Hier definierst du, was das Modell NICHT tun soll, welche Grenzen gelten und welche Annahmen es treffen soll.

Narrowing:

- Budget für alle Maßnahmen zusammen: maximal 200.000€
- Kein zusätzliches Personal (nur vorhandene Ressourcen)
- Fokus auf B2B, nicht B2C
- Keine Blockchain- oder KI-Hype-Vorschläge ohne konkreten ROI
- Branche: Maschinenbau, Mittelstand (200 Mitarbeiter)

RISEN in der Praxis: Vollständiges Beispiel

Role: Du bist ein erfahrener Content-Strategie, der für mittelständische B2B-Unternehmen Content-Marketing-Strategien entwickelt. Du denkst datengetrieben und fokussierst dich auf messbaren ROI.

Instructions: Entwickle eine Content-Strategie für das kommende Quartal für ein Unternehmen, das Industriefilter herstellt und bisher kein Content-Marketing betrieben hat. Die Website hat monatlich 2.000 Besucher, hauptsächlich über Direktzugriffe.

Steps:

1. Analysiere die Ausgangssituation und identifiziere die Zielgruppen (Einkäufer, Ingenieure, Geschäftsführer)
2. Definiere 3-5 Content-Säulen (Themen, die regelmäßig gespielt werden)
3. Erstelle einen Redaktionsplan für 12 Wochen mit konkreten Themen
4. Schlage für jedes Thema das beste Format vor (Blog, Whitepaper, Video, LinkedIn-Post)
5. Definiere KPIs und Meilensteine für die Erfolgsmessung

End Goal: Ein sofort umsetzbarer Redaktionsplan, den ein Marketing-Mitarbeiter ohne Content-Erfahrung abarbeiten kann. Inklusive Erfolgskennzahlen, die nach 3 Monaten ausgewertet werden können.

Narrowing:

- Budget: 3.000€ pro Monat (inkl. freie Texter)
- Kein Video-Equipment vorhanden (nur Smartphone)
- Keine Social-Media-Kanäle außer LinkedIn (Firmenprofil mit 500 Followern)
- Sprache: Deutsch (Zielmarkt DACH)
- Keine Inhalte, die technische Zeichnungen oder geschützte Daten enthalten

Das ist ein langer Prompt. Aber schau dir an, wie klar die Aufgabe definiert ist. Das Modell weiß genau:

- Aus welcher Perspektive es antwortet
- Was es tun soll
- In welcher Reihenfolge
- Wo es ankommen soll
- Welche Grenzen gelten

Die Ergebnisse solcher Prompts sind regelmäßig so gut, dass sie mit minimalem Aufwand direkt verwendbar sind.

RISEN vs. CRAFT: Der direkte Vergleich

Aspekt	CRAFT	RISEN
Fokus	Output-Qualität	Prozess-Steuerung
Stärke	Ton und Format	Schrittfolge und Ziel
Typische Aufgabe	Texte schreiben	Strategien entwickeln
Prompt-Länge	Mittel	Lang
Reproduzierbarkeit	Hoch	Sehr hoch
Einarbeitungszeit	Niedrig	Mittel

Wann du RISEN brauchst – und wann nicht

RISEN ist zu viel für:

- Einfache Texte (E-Mails, kurze Beschreibungen)
- Kreative Aufgaben ohne klaren Prozess
- Schnelle Fragen
- Übersetzungen, Zusammenfassungen

RISEN ist perfekt für:

- Strategien und Pläne
- Analysen mit mehreren Dimensionen
- Aufgaben, die du delegieren würdest und eine klare Briefing-Struktur brauchen
- Alles, wo du denkst: "Ich muss dem Modell erklären, in welcher Reihenfolge es denken soll"

Die Steps sind der Schlüssel

Lass mich den wichtigsten Punkt nochmal betonen: Die Steps machen RISEN einzigartig. Ohne Steps ist RISEN nur ein umständliches CRAFT.

Warum sind Steps so mächtig?

1. **Sie verhindern Abkürzungen.** Ohne Steps springt das Modell gerne direkt zur Lösung, ohne die Situation richtig zu analysieren.
2. **Sie erzwingen Tiefe.** Wenn Step 2 "Identifiziere die 3 größten Herausforderungen" lautet, muss das Modell tatsächlich analysieren, statt sofort Lösungen vorzuschlagen.
3. **Sie strukturieren den Output.** Der Leser (also du) kann die Antwort Step für Step nachvollziehen und prüfen.
4. **Sie ermöglichen Iteration.** Wenn Step 3 nicht passt, kannst du gezielt sagen: "Überarbeite Step 3 mit folgendem Fokus: ..."

Zusammenfassung

- RISEN = Role, Instructions, Steps, End Goal, Narrowing
- Das Framework für komplexe, mehrstufige Aufgaben
- Steps sind das Herzstück – sie steuern den Denkprozess des Modells
- End Goal sorgt für Ergebnisorientierung

- Narrowing verhindert Abschweifen
 - Nicht für einfache Aufgaben – dafür gibt es RTF und CRAFT
-

Übung

RISEN Deep Dive

Wähle eine der folgenden komplexen Aufgaben und erstelle einen vollständigen RISEN-Prompt:

1. **Marktanalyse:** Analysiere den Markt für nachhaltige Verpackungen in Deutschland
2. **Lernplan:** Erstelle einen 8-Wochen-Lernplan für jemanden, der Python lernen will
3. **Event-Planung:** Plane ein Team-Offsite für 25 Personen mit 5.000€ Budget

Wichtig: Die Steps müssen mindestens 4 Punkte umfassen. Das End Goal muss konkret und messbar sein. Und die Narrowing-Elemente müssen mindestens 3 Einschränkungen enthalten.

Teste den Prompt und vergleiche das Ergebnis mit dem, was du bei einem einfachen “Mach mir eine Marktanalyse” (Zero-Shot) bekommen würdest.

Kapitel 8: Frameworks im Vergleich – Welches wann?

Du kennst jetzt drei Frameworks: RTF, CRAFT und RISEN. Die Frage, die sich jeder stellt: Welches soll ich benutzen?

Die Antwort ist einfacher, als du denkst.

Die Entscheidungsmatrix

Frag dich bei jeder Aufgabe drei Fragen:

Frage 1: Wie komplex ist die Aufgabe?

- Einfach → RTF
- Mittel → CRAFT
- Komplex → RISEN

Frage 2: Wie wichtig ist die Qualität beim ersten Versuch?

- Egal, ich überarbeite sowieso → RTF
- Sollte gut sein → CRAFT
- Muss direkt verwendbar sein → RISEN

Frage 3: Gibt es einen klaren Prozess mit mehreren Schritten?

- Nein → RTF oder CRAFT
- Ja → RISEN

Wenn du dir unsicher bist, nimm CRAFT. Es ist der beste Kompromiss.

Alle drei Frameworks an einem Beispiel

Aufgabe: Du willst einen Newsletter-Text über ein neues Produkt schreiben lassen.

RTF-Version

```
Role: Marketing-Texter
Task: Schreibe einen Newsletter-Text, der unser neues
Projektmanagement-Tool "FlowBoard" vorstellt. Hauptfeatu-
res:
KI-gestützte Aufgabenverteilung, Echtzeit-Kollaboration,
Integration mit Slack und Teams.
Format: 200-250 Wörter, mit einem Call-to-Action am Ende
```

Dauer: 30 Sekunden. Ergebnis: Brauchbar, aber generisch.

CRAFT-Version

Context: Wir sind ein B2B-SaaS-Startup mit 500 Newsletter-Abonnenten, hauptsächlich Teamleiter und Projektmanager in Unternehmen mit 20-100 Mitarbeitern. Unser letzter Newsletter hatte eine Öffnungsrate von 35%.

Role: Erfahrener SaaS-Marketing-Texter, der sich auf Launch-Kommunikation spezialisiert hat.

Action: Schreibe einen Newsletter-Text, der unser neues Projektmanagement-Tool "FlowBoard" vorstellt. Hauptfeatures:
KI-gestützte Aufgabenverteilung, Echtzeit-Kollaboration, Integration mit Slack und Teams.

Format: 200-250 Wörter. Struktur: Aufmerksamkeitsstarker Einstieg (Problem, das FlowBoard löst), 3 Features als kurze Absätze, Call-to-Action (kostenlose 14-Tage-Testversion).

Tone: Enthusiastisch aber nicht übertrieben. Nutzen-orientiert statt Feature-orientiert. "Das kann FlowBoard für dich tun" statt "FlowBoard hat diese Features".

Dauer: 2 Minuten. Ergebnis: Deutlich besser. Passt zur Zielgruppe.

RISEN-Version

Role: Senior Content-Strategin mit 8 Jahren Erfahrung in B2B-SaaS-Launches. Du hast Newsletter für Unternehmen wie Asana und Monday geschrieben.

Instructions: Erstelle einen Newsletter-Text für den Launch unseres neuen Projektmanagement-Tools "FlowBoard". Der Newsletter soll informieren und zur Anmeldung für die kostenlose Testversion motivieren.

Steps:

1. Beginne mit einem konkreten Schmerzpunkt, den Projektmanager täglich erleben (z.B. manuelle Aufgabenverteilung, Tool-Chaos)
2. Stelle FlowBoard als Lösung vor (1-2 Sätze, keine Feature-Liste)
3. Hebe die 3 Hauptvorteile hervor (KI-Aufgabenverteilung, Echtzeit-Kollaboration, Slack/Teams-Integration) – jeweils mit einem konkreten Nutzen-Satz
4. Füge einen Social Proof ein (z.B. "Bereits 200 Teams in der Beta-Phase")
5. Schließe mit einem klaren Call-to-Action

End Goal: Der Newsletter soll eine Klickrate von mindestens 5% auf den CTA-Button erzielen. Jeder Leser soll nach dem Lesen genau wissen, was FlowBoard ist und warum er es ausprobieren sollte.

Narrowing:

- Maximal 250 Wörter
- Keine technischen Details (API, Infrastruktur)
- Nicht "revolutionär", "bahnbrechend" oder ähnliche Hyperbeln
- Zielgruppe: Teamleiter, nicht C-Level
- Kein Vergleich mit Wettbewerbern

Dauer: 5 Minuten. Ergebnis: Exzellent. Direkt verwendbar.

Der Zeitaufwand-Qualität-Tradeoff

Ich habe das gerade angedeutet, und es ist der wichtigste Punkt dieses Kapitels:

Framework	Schreibzeit	Ergebnis-Qualität	Nachbearbeitungszeit
RTF	30 Sekunden	60-70%	10-15 Minuten
CRAFT	2-3 Minuten	80-90%	3-5 Minuten
RISEN	5-10 Minuten	90-95%	0-2 Minuten

Wenn du rechnest: RTF braucht insgesamt ~15 Minuten (30 Sek. schreiben + 15 Min. überarbeiten). RISEN braucht insgesamt ~10 Minuten (7 Min. schreiben + 2 Min. überarbeiten). RISEN ist also oft *schneller*, obwohl der Prompt länger dauert.

Aber das gilt nur für komplexe Aufgaben. Für eine einfache Übersetzung wäre ein RISEN-Prompt absurder Overkill.

Frameworks mischen

Noch ein Geheimnis: Du musst dich nicht für ein Framework entscheiden. In der Praxis mische ich ständig.

Mein typischer Workflow:

1. **Anfang:** RTF für die erste Idee
2. **Verfeinerung:** CRAFT für den finalen Prompt

3. **Komplexe Projekte:** RISEN für die Strategie, CRAFT für die Einzeltexte

Das ist kein Betrug. Das ist Pragmatismus.

Andere Frameworks, die du kennen solltest

RTF, CRAFT und RISEN sind nicht die einzigen Frameworks. Hier sind drei weitere, die du vielleicht mal antreffen wirst:

RACE

- **R**ole, **A**ction, **C**ontext, **E**xpectation
- Ähnlich wie CRAFT, mit “Expectation” statt “Tone” und “Format”
- Gut, wenn du klare Erwartungen an das Ergebnis formulieren willst

CRISPE

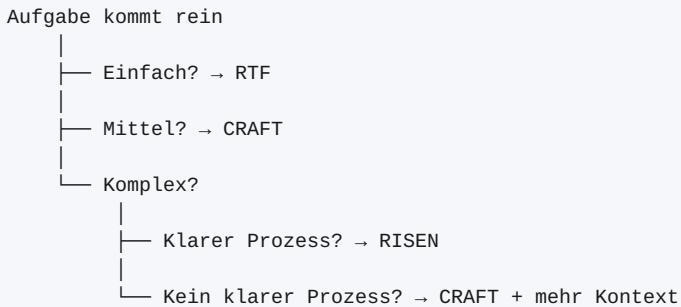
- **C**apacity, **R**ole, **I**nsight, **S**tatement, **P**ersonality, **E**xperiment
- Sechs Elemente – das detaillierteste Framework
- Gut für kreative Aufgaben, wo Persönlichkeit wichtig ist
- Für meinen Geschmack zu umständlich für den Alltag

CO-STAR

- **C**ontext, **O**bjective, **S**tyle, **T**one, **A**udience, **R**esponse
- Sechs Elemente mit starkem Fokus auf Zielgruppe
- Gut für Marketing und Kommunikation

Mein Rat: Lerne RTF, CRAFT und RISEN. Die decken 95% aller Fälle ab. Die anderen Frameworks sind Varianten, die du kennen solltest, aber nicht brauchen wirst.

Die Framework-Entscheidung auf einen Blick



Zusammenfassung

- Es gibt kein “bestes” Framework – nur das richtige für die Aufgabe
 - RTF für Speed, CRAFT für Qualität, RISEN für Komplexität
 - Mehr Aufwand beim Prompt = weniger Aufwand bei der Nachbearbeitung
 - Frameworks können gemischt werden
 - Drei Frameworks reichen für den Alltag
-

Übung

Framework-Battle

Nimm eine Aufgabe deiner Wahl (am besten eine echte aus deinem Arbeitsalltag) und schreibe sie in allen drei Frameworks:

1. RTF-Version
2. CRAFT-Version

3. RISEN-Version

Teste alle drei im selben LLM. Bewerte:

- Welches Ergebnis ist am besten?
- Welcher Prompt hat am längsten gedauert?
- Bei welchem Framework war der Zeitaufwand insgesamt (Prompt + Nachbearbeitung) am niedrigsten?
- Gibt es einen klaren Gewinner?

Dokumentiere deine Ergebnisse. Das ist wertvolles Wissen für deinen persönlichen Prompt-Workflow.

Kapitel 9: Deine persönliche Template-Bibliothek

Du hast jetzt alle Werkzeuge. Shot-Typen, drei Frameworks, Vergleichswissen. Jetzt kommt der Schritt, der dich von einem guten Prompter zu einem effizienten Prompter macht: Du baust dir eine Template-Bibliothek auf.

Was ist eine Template-Bibliothek?

Eine Sammlung von Prompt-Vorlagen, die du immer wieder verwenden kannst. Stell dir ein Kochbuch vor, aber statt Rezepten stehen da deine besten Prompts drin.

Jedes Template ist ein erprobter Prompt, bei dem du nur noch die variablen Teile ausfüllen musst. Alles andere – Rolle, Format, Ton, Struktur – ist fest definiert und bewährt.

Warum Templates der entscheidende Vorteil sind

Drei Gründe:

1. Zeitersparnis

Ohne Template schreibst du jeden Prompt von Grund auf. Das dauert bei CRAFT 2-3 Minuten, bei RISEN 5-10 Minuten. Bei 20 Prompts am Tag ist das eine Stunde nur für Prompt-Schreiben.

Mit Templates füllst du nur die Lücken aus. Das dauert 30 Sekunden. Rechne das hoch auf eine Woche.

2. Konsistente Qualität

Deine Templates basieren auf Prompts, die funktioniert haben. Du weißt, dass sie gute Ergebnisse liefern. Ohne Template improvisierst du jedes Mal – und mal ist es gut, mal weniger.

3. Wissenstransfer

Wenn du deine Templates dokumentierst, können auch Kollegen oder Teammitglieder sie nutzen. Das ist besonders in Unternehmen Gold wert, wo verschiedene Leute mit KI arbeiten, aber nicht alle gleich erfahren sind.

So baust du deine Bibliothek auf

Schritt 1: Sammle deine wiederkehrenden Aufgaben

Geh eine typische Arbeitswoche durch und notiere jede Aufgabe, bei der du ein LLM benutzt oder benutzen könntest:

- E-Mails schreiben
- Texte zusammenfassen
- Präsentationen erstellen
- Berichte formulieren
- Social-Media-Posts erstellen

- Code überprüfen
- Brainstorming
- Recherche
- Übersetzungen
- Feedback formulieren

Schritt 2: Priorisiere

Welche Aufgaben machst du am häufigsten? Die kommen zuerst. Du brauchst nicht 50 Templates. Fang mit 5-10 an.

Schritt 3: Wähle das passende Framework

Für jede Aufgabe: Ist RTF genug? Braucht es CRAFT? Oder RISEN? Nutze die Entscheidungsmatrix aus Kapitel 8.

Schritt 4: Schreibe das Template

Jedes Template hat feste und variable Teile. Die festen Teile bleiben immer gleich. Die variablen Teile markierst du mit Platzhaltern.

Schritt 5: Teste und verbessere

Benutze jedes Template mindestens 3-5 Mal, bevor du es als "fertig" betrachtest. Passe es nach jedem Einsatz an, wenn nötig.

Template-Beispiele

Template 1: Professionelle E-Mail (RTF)

Role: Professioneller Business-Kommunikator mit Erfahrung in [BRANCHE]

Task: Schreibe eine E-Mail an [EMPFÄNGER] zum Thema [THEMA].
Kernbotschaft: [KERNBOTSCHAFT].
Gewünschte Reaktion: [WAS SOLL DER EMPFÄNGER TUN?]

Format: Betreffzeile + E-Mail-Text. Maximal [LÄNGE] Wörter. [ANZAHL] Absätze.

Anwendung:

Role: Professioneller Business-Kommunikator mit Erfahrung in der IT-Branche

Task: Schreibe eine E-Mail an den Projektleiter zum Thema Budgetüberschreitung. Kernbotschaft: Wir brauchen 15% mehr Budget für Phase 3. Gewünschte Reaktion: Termin für Budget-Meeting vereinbaren.

Format: Betreffzeile + E-Mail-Text. Maximal 150 Wörter. 3 Absätze.

Template 2: Blog-Artikel (CRAFT)

Context: Blog über [THEMA] für [ZIELGRUPPE]. [ZUSÄTZLICHER KONTEXT].

Role: [EXPERTEN-ROLLE] mit [ERFAHRUNG/SPEZIALISIERUNG].

Action: Schreibe einen Blogartikel über [ARTIKELTHEMA].
Hauptpunkte: [PUNKT 1], [PUNKT 2], [PUNKT 3].

Format: [WORTANZAHL] Wörter. Struktur: Einleitung (Hook + Problem), [ANZAHL] Hauptteile mit Zwischenüberschriften, Fazit mit [CTA/ZUSAMMENFASSUNG]. Verwende [FORMATIERUNGSWÜNSCHE].

Tone: [TONBESCHREIBUNG]. [STIL-DETAILS].

Template 3: Strategische Analyse (RISEN)

Role: [EXPERTEN-ROLLE] mit Spezialisierung auf [FACHGEBIET].

Instructions: Analysiere [ANALYSEOBJEKT] und entwickle [GEWÜNSCHTES ERGEBNIS]. Berücksichtige [RAHMENBEDINGUNGEN].

Steps:

1. Fasse die Ausgangssituation zusammen
2. Identifiziere die [ANZAHL] wichtigsten [CHANCEN/PROBLEME/TRENDS]
3. Entwickle für jede/s [CHANCE/PROBLEM/TREND] konkrete Handlungsempfehlungen
4. Priorisiere nach [KRITERIEN]
5. Erstelle einen Maßnahmenplan mit Zeitrahmen

End Goal: [KONKRETES ERGEBNIS], das [WER] für [ZWECK] verwenden kann.

Narrowing:

- [EINSCHRÄNKUNG 1]
- [EINSCHRÄNKUNG 2]
- [EINSCHRÄNKUNG 3]

Template 4: Social-Media-Post (CRAFT + One-Shot)

Context: [PLATTFORM]-Account für [UNTERNEHMEN/PERSON].
Zielgruppe: [ZIELGRUPPE]. Bisherige Tonalität: [STIL].

Role: Social-Media-Manager mit Erfahrung in [BRANCHE].

Action: Erstelle [ANZAHL] Posts zum Thema [THEMA].

Format: Jeweils: Post-Text (max. [ZEICHENLIMIT] Zeichen) +
Hashtag-Vorschläge (max. [ANZAHL]) + Bild-Beschreibung
(1 Satz).

Tone: [TONBESCHREIBUNG].

Beispiel für unseren Stil:
[BEISPIEL-POST EINFÜGEN]

Template 5: Feedback geben (RTF)

Role: Empathischer aber ehrlicher Kommunikationsberater

Task: Formuliere konstruktives Feedback für [PERSON/ROLLE]
zum

Thema [FEEDBACK-THEMA]. Positiv: [WAS GUT WAR]. Verbesserung:
[WAS BESSER WERDEN SOLL]. Konkreter Vorschlag: [IDEE].

Format: 3 Absätze (Positives – Verbesserungspotenzial –
nächste Schritte). Maximal [LÄNGE] Wörter. Direkte Ansprache.

Wo speicherst du deine Templates?

Es gibt keine perfekte Lösung, aber hier sind die gängigsten:

Option 1: Notion / Obsidian / Evernote

Flexibel, durchsuchbar, mit Tags organisierbar. Mein persönlicher Favorit ist Notion, weil ich Templates in Datenbanken organisieren kann.

Option 2: Google Docs / Word

Einfach, überall zugänglich. Ein Dokument pro Kategorie (Marketing, Kommunikation, Analyse, etc.)

Option 3: GitHub / GitLab

Für die Technisch-Affinen. Versionierung inklusive – du siehst, wie sich deine Templates über die Zeit verbessert haben.

Option 4: Direkt im LLM

Manche Tools wie ChatGPT bieten “Custom Instructions” oder gespeicherte Prompts. Praktisch, aber an ein Tool gebunden.

Template-Organisation

Egal wo du speicherst – organisiere deine Templates so, dass du sie in unter 10 Sekunden findest:

Kategorien (Vorschlag):

- Kommunikation (E-Mails, Nachrichten, Feedback)
- Content (Blog, Social Media, Newsletter)
- Analyse (Marktanalyse, Wettbewerb, Daten)
- Kreativ (Brainstorming, Texte, Ideen)
- Technisch (Code, Dokumentation, Debugging)

Für jedes Template dokumentiere:

- Name (kurz und beschreibend)
- Kategorie

- Framework (RTF/CRAFT/RISEN)
- Wann verwenden (1 Satz)
- Letzte Änderung
- Bewertung (funktioniert gut / muss verbessert werden)

Die 80/20-Regel der Templates

Du wirst feststellen: 20% deiner Templates benutzt du 80% der Zeit. Das ist normal. Konzentriere deine Energie auf diese 20%. Mach sie perfekt. Die anderen Templates? Gut genug reicht.

Und noch was: Templates sind lebende Dokumente. Wenn ein Prompt nicht mehr funktioniert – weil das LLM aktualisiert wurde, weil sich deine Anforderungen geändert haben, weil du was Besseres gefunden hast –, dann aktualisiere das Template. Lösche alte, die du nicht mehr brauchst.

Zusammenfassung

- Eine Template-Bibliothek spart Zeit und sichert Qualität
 - Starte mit 5-10 Templates für deine häufigsten Aufgaben
 - Markiere variable Teile als Platzhalter [IN GROSSBUCHSTABEN]
 - Teste jedes Template 3-5 Mal bevor du es als fertig betrachtest
 - Organisiere nach Kategorien und halte Templates aktuell
 - 20% deiner Templates decken 80% deiner Aufgaben ab
-

Übung

Dein Starter-Kit

Erstelle deine ersten 5 Templates:

1. Identifiziere 5 Aufgaben, die du regelmäßig mit KI erledigst (oder erledigen könntest)
2. Wähle für jede Aufgabe das passende Framework
3. Schreibe das Template mit Platzhaltern
4. Teste jedes Template einmal
5. Speichere sie an einem Ort, wo du sie schnell wiederfindest

Du hast damit den Grundstein für dein persönliches Prompt-System gelegt. Ab jetzt baust du es Stück für Stück aus.

Kapitel 10: Zusammenfassung und Ausblick

Zwei Bände. Du bist jetzt offiziell kein Anfänger mehr.

Lass mich zusammenfassen, was du in Band 2 gelernt hast – und wohin es in Band 3 geht.

Was du jetzt kannst

1. **Shot-Typen anwenden** – Du weißt, wann Zero-Shot reicht, wann ein Beispiel hilft (One-Shot) und wann du mehrere brauchst (Few-Shot).
2. **CRAFT nutzen** – Context, Role, Action, Format, Tone. Dein All-rounder-Framework für die meisten Aufgaben.
3. **RTF für Speed** – Role, Task, Format. Drei Elemente, 30 Sekunden, brauchbares Ergebnis.
4. **RISEN für Komplexität** – Role, Instructions, Steps, End Goal, Narrowing. Das Framework, wenn es wirklich drauf ankommt.
5. **Frameworks vergleichen und wählen** – Du hast eine Entscheidungsmatrix und weißt, wann welches Framework passt.
6. **Templates erstellen** – Du kannst Prompt-Vorlagen bauen, die du immer wieder verwenden kannst.

Checkliste: Bin ich bereit für Band 3?

- Ich kann Zero-Shot, One-Shot und Few-Shot erklären und anwenden
- Ich habe CRAFT mindestens 5 Mal in der Praxis benutzt
- Ich kenne den Unterschied zwischen RTF, CRAFT und RISEN
- Ich habe mindestens einen RISEN-Prompt geschrieben und getestet
- Ich habe angefangen, eine Template-Bibliothek aufzubauen
- Ich kann entscheiden, welches Framework für eine Aufgabe passt
- Ich habe Few-Shot mit mindestens 3 Beispielen ausprobiert
- Ich verstehe, warum die Qualität der Beispiele entscheidend ist
- Ich habe mindestens 3 Templates erstellt und getestet
- Ich merke, dass meine Prompts besser werden

Bei 7 von 10? Weiter zu Band 3.

Was dich in Band 3 erwartet

Band 3 heißt “Fortgeschrittene Basics” – und ja, das klingt wie ein Widerspruch. Ist es aber nicht. Es sind Techniken, die über die Grundlagen hinausgehen, aber noch nicht in die Experten-Ecke gehören.

Konkret:

Prompt-Chaining

Statt einem einzelnen Mega-Prompt: Eine Kette von Prompts, bei der jeder auf dem Ergebnis des vorherigen aufbaut. Das Ergebnis ist besser als alles, was ein einzelner Prompt schaffen könnte.

Delimiter und Strukturierung

Wie du mit Trennzeichen (`---`, `###`, `"""`) deine Prompts so strukturierst, dass das Modell verschiedene Teile klar unterscheiden kann. Besonders wichtig bei langen Prompts.

Negative Prompts

Was das Modell NICHT tun soll. Klingt simpel, ist aber eine eigene Kunst. “Kein Marketing-Sprech” ist ein negatives Prompt. Und manchmal ist es effektiver als zu sagen, was du willst.

Temperatur und Parameter

In Band 1 habe ich Temperatur kurz erklärt. In Band 3 gehen wir tiefer: Wie du Temperatur, Top-P und andere Parameter bewusst einsetzt – und warum das nur über die API geht.

System-Prompts

Das mächtigste Werkzeug, das die meisten Leute nicht kennen. System-Prompts definieren das Grundverhalten des Modells und werden vor deinem eigentlichen Prompt ausgeführt. Wir schauen uns an, was das bedeutet und wie du es nutzen kannst.

Wie die Reihe weitergeht

Zur Erinnerung:

Anfänger (Band 1-3)

- Band 1: Grundlagen ✓
- Band 2: Prompt-Frameworks ✓ *(du bist hier)*
- Band 3: Fortgeschrittene Basics

Fortgeschritten (Band 4-6)

- Band 4: Reasoning-Techniken
- Band 5: Kreatives Prompting
- Band 6: Spezialisiertes Prompting

Profi (Band 7-9)

- Band 7: Prompting für Entwickler
- Band 8: Business & Produktivität
- Band 9: Sicherheit & Ethik

Experte (Band 10)

- Band 10: Die Zukunft

Ein Gedanke zum Schluss

Frameworks sind Werkzeuge. Nicht mehr, nicht weniger. Sie helfen dir, systematisch zu denken und konsistente Ergebnisse zu produzieren. Aber sie ersetzen nicht dein eigenes Urteil.

Der beste Prompt der Welt nutzt nichts, wenn du nicht weißt, ob das Ergebnis gut ist. Wenn du nicht merkst, dass die KI Unsinn erzählt. Wenn du nicht die Erfahrung hast, das Ergebnis einzuordnen.

Frameworks geben dir Struktur. Erfahrung gibt dir Qualität. Und Erfahrung bekommst du nur durch eines: Machen.

Also mach. Prompt. Experimentiere. Feile an deinen Templates. Und wenn du bereit bist, schnapp dir Band 3.

Wir sehen uns dort.

Belkis Aslani

Ressourcen und weiterführende Links

Frameworks vertiefen:

- Prompt Engineering Guide (promptingguide.ai) – Shot-Typen im Detail
- Learn Prompting (learnprompting.org) – Interaktive Framework-Übungen
- DAIR.AI Prompt Engineering Guide – Akademisch fundiert

Templates finden:

- PromptBase – Marktplatz für Prompt-Templates
- awesome-chatgpt-prompts (GitHub) – Kostenlose Community-Prompts
- FlowGPT – Prompt-Sharing-Plattform

Zum Ausprobieren:

- ChatGPT Custom Instructions – Für gespeicherte System-Prompts
- Claude Projects – Für projektbezogene Prompt-Sammlungen
- Google AI Studio – Für Parameter-Experimente (Temperatur, Top-P)

Wissenschaftlich:

- “Language Models are Few-Shot Learners” (Brown et al., 2020) – Das Paper, das Few-Shot Prompting populär gemacht hat
- “A Survey on In-context Learning” (Dong et al., 2024) – Umfassende Übersicht über Shot-Typen und ihre Wirkung