

• RTK DEEP DIVE • WORKSHOP

KI im *Projektmanagement*

Planung und Umsetzung mit KI-Unterstützung: vom Prompt zum System.

Gustavo Salami · Kuble AG · [Downloads für Teilnehmende](#) ↓

• MINDSET

Vor dem Prompt kommt *Kontext*. Aus dem Kontext wird ein *System*.

GESTERN

Prompting

Jede Frage einzeln formulieren. Kontext jedes Mal neu zusammensuchen.

HEUTE

Context Engineering

Der Kontext liegt strukturiert bereit. Die KI holt sich, was sie braucht.

MORGEN

Loop Engineering

Systeme laufen selbstständig. Der Mensch prüft und entscheidet, statt jeden Schritt zu steuern.

Ich schreibe kaum noch Prompts. Ich sage der KI: «Mach mir einen Prompt.» Wenn der Kontext stimmt, stimmt auch der Prompt.

Was die KI übernimmt, was bei dir bleibt

Die KI kann

- E-Mails und Threads entwirren und zusammenfassen
- Statusupdates, To-dos und Timelines pflegen
- Briefing-Chaos strukturieren, Widersprüche finden
- Verdichten, Varianten schreiben, Risiken markieren
- Ausführlicher und geduldiger arbeiten als jeder Mensch unter Zeitdruck

Der Mensch muss

- Entscheidungen fällen: die Verantwortung bleibt bei dir
- Involviert sein, sonst kannst du nichts gegenchecken
- Ergebnisse prüfen und verifizieren
- Wissen, was er verstehen muss und was nicht

«Verstehen kannst du nicht delegieren.»

• DIE GRÖSSTE GEFAHR

Weniger ist *mehr*

Die KI baut dir alles, auch Systeme, die du selber nicht mehr durchschaust. Nach einer Woche weisst du nicht mehr, was das Ding überhaupt macht. Und die Verantwortung bleibt trotzdem bei dir.

Zu wissen, was du *nicht* brauchst, ist wichtiger als zu wissen, was du willst.

Herunterskalieren ist schwieriger als komplex bauen. Ein minimales System, das du verstehst, schlägt jedes Wunderwerk.

Wenige Automatisierungen, die du im Griff hast, statt tausend Dinge, die irgendwann irgendetwas verschicken.

• TEIL 1 • FUNDAMENT

Ground *Truth*

Kontext, dem du trauen kannst.

Ground Truth

Ein strukturierter, kuratierter Ort, an dem das Kernwissen zum Projekt liegt: Ziele, Scope, Stakeholder, Aufgaben, Entscheidungen, Risiken. Nicht ein Ordner, in dem einfach alles drinsteckt.

Warum Markdown, nicht Word

- Reiner Text mit einfacher Struktur: KI und Mensch lesen dasselbe
- Kein Formatierungsmüll, keine Binärdaten, keine «Verunreinigung»
- Die KI kann daraus jederzeit Website, PDF oder Präsentation machen

So sieht das aus

```
# Projekt Nova Pulse
## Status
- Gelb: Budget reduziert auf CHF 42'000

## Nächster Schritt
- GL-Entscheidungsgrundlage bis 10:30

## Offene Fragen
- Bern-Sales bestätigt?
```

Kanonisch heisst: die eine, verbindliche Fassung. Widersprechen sich zwei Angaben, gilt, was in dieser Datei steht. Rohquellen (Mails, Briefings, Transkripte) bleiben daneben liegen, die Essenz wird hier kuratiert.

Drei Bausteine, dann fühlt es sich an wie Assistenz

1

Ground Truth

Das kuratierte Projektwissen. Lokal, in Markdown, von der KI gepflegt.

2

Verbindungen

MCP, Connectoren, Plugins, vor allem zur E-Mail. Dort läuft die Projektkommunikation. Gmail-Labels oder Outlook-Ordner statt der ganzen Inbox.

3

Trigger & Automatisierung

Ein Wort statt ein Prompt: `status` , `update` , `check` . Und jeden Morgen läuft das Briefing von selbst.

«Eine gute Assistenz denkt mit und weiss jeden Morgen, was ansteht. Genau das kann die KI übernehmen, und sie schaut auch am Freitagabend kein Fussballspiel.»

• TEIL 2 • LIVE-DEMO

Lumen & *Laut*

Eine fiktive Agentur, ein echtes System.

So läuft die Demo

- 1 **Der Ordner:** Quellen in `sources/`, eine Datei `ground-truth.md`, alles lokal.
- 2 **Das Dashboard:** läuft auf localhost, nur bei mir. Zwei Ansichten aus derselben Datei: eine komplexe und eine minimalistische. Gleiche Wahrheit, andere Darstellung.
- 3 **Trigger:** Ich schreibe `status`, die KI liest die Ground Truth und sagt mir, wo das Projekt steht und was die nächste beste Aktion ist.
- 4 **E-Mail:** Neue Kundinnenmail im Projeklabel → die KI erstellt einen *Draft* direkt in Gmail. Ich prüfe und schicke ab.
- 5 **Artefakte:** «Mach mir eine Timeline» → Markdown → «Mach eine Website daraus» → «Und ein PDF für die GL.» Alles aus demselben Kontext.
- 6 **Meeting-Transkript:** hineinwerfen → Entscheidungen, Aufgaben mit Owner, offene Fragen, sauber extrahiert.

Wichtig fürs Publikum: Das Dashboard ist nur eine Spiegelung der Ground Truth. Ich klicke dort nichts, ich sage dem Agenten, was zu ändern ist.

E-Mail-Zugang, ohne alles preiszugeben

Labels statt Inbox. Die KI liest nur Mails mit dem Projektlabel (Gmail) oder aus dem Projektordner (Outlook), nie die ganze Inbox.

Eigene Agent-Inbox als Wand. Der Agent bekommt eine eigene E-Mail-Adresse. Projektmails leitest du weiter, sensible Mails nicht. Fertig ist die Trennung.

Drafts statt Auto-Versand. Die KI entwirft Antworten, du gibst frei. Automatisch senden lassen würde ich nicht.

Schweizer Optionen. Open-Weight-Modelle (Kimi, GLM) lassen sich in der Schweiz hosten, z.B. bei Infomaniak. Mistral hat Connectoren und sitzt in Europa.

Nutzen gegen Risiko abwägen: Was ist das Schlimmste, das passieren kann? Das ist deine Einschätzung, und deine Verantwortung.

• TEIL 3 · FÜR DICH

Zum *Mitnehmen*

Prompts, Starter-Kit und der Einstieg ohne Vollzugriff.

Use Cases & Prompts: *kopieren und loslegen*

Funktionieren in ChatGPT, Claude und Copilot, solange der Kontext mitkommt. Jeder gute PM-Prompt hat fünf Teile: Quelle, Auftrag, Format, Review-Regel, nächster Schritt.

Der wichtigste Zusatz, an jeden Prompt anhängen

Trenne immer:

- belegte Fakten
- Annahmen
- offene Fragen

Erfinde keine Fakten. Wenn Informationen fehlen, sag genau, welche.

Briefing-Chaos strukturieren

Hier sind mehrere Briefing-Versionen, Notizen und Mails zum selben Projekt.

Hilf mir, eine Struktur hineinzubringen:

- Was ist der aktuelle, gültige Stand?
- Wo widersprechen sich die Versionen?
- Stimmen Timelines und Budgets überein?
- Was hat sich über die Zeit verändert?

Erstelle danach eine konsolidierte Fassung mit offenen Fragen an den Kunden.

Meeting-Transkript → Protokoll & Aufgaben

Nutze dieses Transkript.

Extrahiere:

- Entscheidungen
- Aufgaben mit Owner und Deadline
- offene Fragen
- Risiken
- Punkte fürs Follow-up

Gib danach eine Kund:innenmail als Entwurf aus.

Keine Aufgabe ohne Owner als Fakt behandeln.

Unsichere Deadlines als offene Frage markieren.

E-Mail-Thread entwirren

Hier ist ein E-Mail-Thread mit mehreren Beteiligten und vermischten Themen.

- Fasse zusammen, was inhaltlich wirklich passiert ist.
- Wer will was von wem, und bis wann?
- Wo widerspricht sich etwas oder bezieht sich jemand auf Veraltetes?
- Was muss ich beantworten, was kann ich ignorieren?

Erstelle danach einen Antwortentwurf in freundlichem, klarem Ton.

Statusbericht

Nutze ausschliesslich die Projekt-Ground-Truth und die neuen Quellen.

Erstelle einen Statusbericht:

1. Kurzfazit
2. Fortschritt seit letztem Update
3. Blocker
4. Risiken
5. Entscheidungen nötig
6. Nächste Schritte
7. Offene Fragen

Markiere pro Aussage die Quelle oder «Annahme».

Stakeholder-Update in 3 Versionen

Nutze den aktuellen Projektstand.

Erstelle drei Versionen desselben Updates:

1. Management-Summary
2. Team-Update
3. Kund:innenmail

Gleiche Fakten, aber Ton, Detailtiefe und Call-to-Action angepasst.

Gib am Schluss aus, was vor dem Versand geprüft werden muss.

Risiko-Review

Prüfe diesen Projektstand auf Risiken.

Liefere die Top-5-Risiken mit:

- Wahrscheinlichkeit und Schaden
- Frühwarnsignal
- Gegenmassnahme
- Owner-Vorschlag
- Quelle oder Annahme

Markiere Risiken, die aus fehlenden Informationen entstehen.

Ground Truth aktualisieren

Nutze die neuen Quellen in [Ordner oder Label].

Aktualisiere die Ground Truth:

1. Neue Quellen auflisten
2. Fakten extrahieren
3. Annahmen und offene Fragen separat markieren
4. Aufgaben, Risiken und Entscheidungen aktualisieren
5. Changelog schreiben

Keine alten Fakten überschreiben, ohne den Grund im Changelog zu dokumentieren.

Timeline & Meilensteine

Erstelle aus der Ground Truth eine Timeline mit den wichtigsten Meilensteinen, Abhängigkeiten und Verantwortlichen.

Prüfe deinen eigenen Plan danach kritisch:

- Was ist zu optimistisch?
- Wo fehlen Quellen?
- Welche Entscheidung braucht menschliche Freigabe?

Danach: Stelle die Timeline als übersichtliche Tabelle dar.

- UND WENN ICH NUR COPILOT HABE?

Nur Microsoft Copilot? Damit arbeitest du *trotzdem*

Ohne Datei-Agent und Connectoren bist du selber die Verbindung: du trägst den Kontext in den Chat und die Ergebnisse zurück. Mehr Copy-Paste, gleiches Prinzip.

1 **Immer «Think Deeper» bzw. Reasoning an**, sobald etwas kombiniert werden muss. Faustregel: Könnte es ein Praktikant nicht erledigen, dann Think Deeper. Nur beim Spellcheck lohnt sich der schnelle Modus.

2 **Du bist der Connector.** Mails, Notizen und Protokolle bewusst in den Chat geben, Ergebnisse zurück in deine Ablage kopieren. Das ist der manuelle Loop, und er funktioniert.

3 **E-Mails als .eml aus Outlook ziehen** und in den Chat hochladen. Copilot liest sie direkt, kein PDF-Ausdrucken, kein Umformatieren. Zehn wichtige Mails reichen für den Anfang.

4 **Copilot dort nutzen, wo es eingebaut ist:** in Outlook Threads zusammenfassen und Antworten entwerfen, in Teams das Meeting-Recap, in Word und Excel direkt im Dokument.

5 **Die IT fragen**, was freigegeben ist: SharePoint-Zugriff? Outlook-Connector? Schlag ein Pilotprojekt vor, statt auf die grosse Freigabe zu warten.

Dein GTS lebt im *SharePoint*

Das Ground-Truth-Prinzip braucht keinen lokalen Agenten. Eine gepflegte Datei im Projektordner reicht, du übernimmst die Rolle der Automatisierung.

1 **Eine Ground-Truth-Datei anlegen** im SharePoint- oder OneDrive-Projektordner. Ideal als Textdatei (.md oder .txt), die Vorlage dafür ist im GTS-Starter-Kit. Wenn es Word sein muss: schlicht halten, Titel und Listen statt Tabellenwüsten.

2 **In jedem Chat mitgeben:** Datei anhängen oder (mit M365 Copilot, je nach Lizenz) direkt referenzieren. Copilot hat damit sofort den ganzen Projektkontext, egal in welchem Chat du bist.

3 **Update-Routine etablieren:** neue Mails und Protokolle hochladen, «Aktualisiere die Ground Truth» ausführen, das Ergebnis zurück in die Datei speichern. Fünf Minuten am Morgen, dafür stimmt der Stand.

4 **In Steps arbeiten:** zuerst verstehen lassen, dann extrahieren, dann erstellen. Nach jedem Schritt prüfen, denn die Verantwortung bleibt bei dir.

Erster Prompt für den Einstieg

Ich lade dir die wichtigsten E-Mails, das Briefing und ein Meeting-Protokoll zu einem Projekt hoch.

Hilf mir zuerst zu verstehen, was hier läuft. Danach extrahiere das Wissen so, dass ich es als Projekt-Grundlage (Ground Truth) weiterverwenden kann:

Ziele, Scope, Beteiligte, Aufgaben, Entscheidungen, Risiken, offene Fragen.

Trenne belegte Fakten, Annahmen und offene Fragen. Erfinde nichts.

Update-Routine für Copilot

Angehängt findest du meine Ground-Truth-Datei und neue Quellen (Mails, Protokolle) seit dem letzten Update.

Aktualisiere die Ground Truth:

1. Fakten aus den neuen Quellen extrahieren
2. Aufgaben, Risiken und Entscheidungen nachführen
3. Annahmen und offene Fragen separat markieren
4. Änderungen am Schluss als kurze Liste ausgeben

Gib mir die komplette aktualisierte Datei zurück, damit ich sie direkt wieder abspeichern kann. Erfinde nichts.

Alles zum Herunterladen und Sofort-Loslegen



Prompt-Bibliothek

10 Kern-Prompts nach Use Case, Trigger-Prompts für ein GTS, Copilot-Leitfaden, Beispiel mit Meeting-Transkript.

[prompt-bibliothek.zip](#) ↓



GTS-Starter-Kit

Ground-Truth-Vorlage, Agenten-Regeln, Ordnerstruktur und der Start-Prompt, der alles aufbaut.

[gts-starter-kit.zip](#) ↓



Übungsdaten

Synthetisches Briefing, Mails und Mediaplan der fiktiven Agentur, plus 4 Übungen zum Ausprobieren.

[uebungsdaten.zip](#) ↓

Die Merksätze

- Vor dem Prompt kommt Kontext. Aus dem Kontext wird ein System.
- Verstehen kannst du nicht delegieren, die Verantwortung bleibt bei dir.
- Weniger ist mehr: Wissen, was du *nicht* brauchst.
- Eine kanonische Markdown-Datei schlägt jeden Dokumenten-Friedhof.
- KI muss man kennenlernen wie einen Menschen. Anfangen, prüfen, Vertrauen aufbauen.

Mehr davon: [rtk.kuble.com](https://www.rtk.kuble.com), Road to Knowledge: gratis registrieren, Tutorials und regelmässige Webinare.

In 6 Schritten zu deinem eigenen *PM-System*

Die meisten holen sich von der KI Antworten. Du baust dir ein System. Sechs Schritte, jeder ist für sich schon nützlich.

1 **ueber-mich.md schreiben.** Wer du bist, deine Rolle, deine Projekte, dein Kommunikationsstil, deine No-Gos. Das ist das Fundament: besserer Kontext, bessere Resultate.

2 **AGENTS.md anlegen.** Die Betriebsanleitung für deine KI: wie sie denkt und schreibt, wann sie nachfragt, was tabu ist. Fakten von Annahmen trennen, Drafts statt Versand. Du hörst auf, dich zu wiederholen.

3 **Ground Truth pro Projekt aufbauen.** Eine kanonische Datei je Projekt statt tausend Chats: die eine, verbindliche Fassung, bei Widersprüchen gilt sie. Rohquellen daneben in `sources/`. Das ist das Gedächtnis deines Systems.

4 **Verbindungen einrichten.** E-Mail über Projektlabels, Kalender, Ablage. Erst mit Signalen von aussen wird aus dem Chat eine Assistenz.

5 **Trigger statt Prompts.** `update`, `check`, `status`. Einmal in der AGENTS.md definiert, für immer wiederverwendbar.

6 **Automatisieren.** Morgen-Briefing, Statusreport, Deadline-Wächter. Dein System arbeitet auch, wenn du nicht hinschaust.

Die Prompts dazu: kopieren, einfügen, und die KI interviewt dich. Am Schluss bekommst du die Datei fertig formatiert zum Abspeichern.

1 · ueber-mich.md erstellen

Ich will eine Datei ueber-mich.md als Fundament für meine KI-Zusammenarbeit.

Interviewe mich dazu: Stelle mir die Fragen einzeln und warte jeweils auf meine Antwort. Frage nach: Rolle und Verantwortung, aktuelle Projekte, Arbeitsweise, Kommunikationsstil (Ton, Sprache, Du/Sie), Werte und Entscheidungsgrundsätze, No-Gos.

Maximal 8 Fragen. Bohre nach, wenn eine Antwort zu vage ist.

Danach erstelle die komplette ueber-mich.md als Markdown mit klaren Abschnitten (# Titel, ## Abschnitte, Aufzählungen), damit ich sie direkt abspeichern kann. Nutze nur meine Antworten, erfinde nichts.

2 · AGENTS.md erstellen

Ich will eine AGENTS.md: die Betriebsanleitung, wie meine KI für mich arbeiten soll.

Interviewe mich einzeln zu: Wie soll die KI schreiben (Ton, Sprache, Länge)? Was soll sie nie tun? Wann soll sie nachfragen statt raten? Was darf sie allein entscheiden, was braucht meine Freigabe? Welche Triggerwörter will ich (z.B. update, check, status) und was soll jeweils passieren?

Danach erstelle die komplette AGENTS.md als Markdown: Grundregeln, Arbeitsweise, Tabus, Triggerwörter mit Ablauf. Formuliere die Regeln kurz und prüfbar. Nur aus meinen Antworten, nichts erfinden.

3 · Ground Truth aufbauen

Ich will für ein Projekt eine Ground Truth aufbauen: eine kanonische Datei, die den verbindlichen Stand hält.

Interviewe mich einzeln zu: Projektname und Ziel, Scope und Out-of-Scope, Beteiligte und Rollen, aktueller Status, nächste Schritte mit Fristen, bekannte Risiken, offene Fragen.

Danach erstelle die komplette ground-truth.md als Markdown mit den Abschnitten: Zusammenfassung, Projekt, Aufgaben (Tabelle mit Owner und Frist), Entscheidungen, Risiken, Offene Fragen, Annahmen, Changelog. Trenne belegte Fakten von Annahmen. Erfinde nichts.

4 · Verbindungen planen

Hilf mir zu entscheiden, welche Verbindungen meine KI zu meinen Tools bekommen soll.

Frage mich einzeln: Welche Tools nutze ich (E-Mail, Kalender, Ablage, Chat)? Was davon darf die KI lesen, was schreiben? Welche Daten sind sensibel und bleiben aussen vor? Was erlaubt meine Firma?

Danach erstelle mir einen kurzen Verbindungs-Plan als Markdown: pro Tool der Zugriff (lesen / schreiben / kein Zugriff), die Begründung und der erste Einrichtungsschritt. Dazu eine Zeile pro Risiko, das ich bewusst in Kauf nehme.

5 · Trigger definieren

Ich will Triggerwörter definieren, damit ich nicht jedes Mal neu prompten muss.

Frage mich einzeln: Welche Abläufe wiederhole ich regelmässig (z.B. Tagesstand, neue Quellen verarbeiten, Statusreport)? Was genau soll bei jedem Ablauf passieren, Schritt für Schritt? Was soll die KI dabei nie tun?

Danach erstelle mir pro Ablauf einen Trigger-Abschnitt im AGENTS.md-Format: ### triggerwort, darunter die Schritte als nummerierte Liste, darunter die Grenzen. Kurz und eindeutig, damit die KI es jedes Mal gleich ausführt.

6 · Automatisierung aufsetzen

Ich will wiederkehrende PM-Arbeit automatisieren (z.B. Morgen-Briefing, Statusreport, Deadline-Wächter).

Frage mich einzeln: Welche Routine will ich zuerst? Wann soll sie laufen? Welche Quellen braucht sie? Was ist das Ergebnis und wohin geht es? Woran merke ich, dass sie falsch läuft?

Danach erstelle mir die Automatisierungs-Anweisung als Markdown: Name, Zeitplan, Schritte, Quellen, Ausgabeformat, Stopp-Regeln. So formuliert, dass ich sie direkt als Auftrag an meine KI oder in eine Automation einfügen kann. Starte mit einer Routine, nicht mit fünf.

«Die meisten stellen der KI Fragen. Wenige bauen Systeme. Das ist der Unterschied.»