Spielleiter-Liste: 40 Fragen mit Auflösung

40 Fragen in fünf Farben, je 8 pro Farbe. Die Reihenfolge ergibt sich aus dem gemischten Stapel der Fragenkarten; hier hakst du ab, was schon gefragt wurde. Timer: 30 Sekunden. Die fett gesetzte Option ist die richtige. ★ markiert Schlüsselfragen, bei denen die Gesprächsrunde empfohlen ist.

Blau · Modell

☐ **Blau 1** — Was ist ein Token?
A Immer genau ein Wort · B Ein ganzer Satz · **C Ein Wortstück: oft ein Wort, manchmal nur ein Teil**
Häufige Wörter bleiben ein einziges Stück, seltene zerfallen in mehrere Teile, bis hinunter zum Buchstaben. Ein Token ist deshalb nicht dasselbe wie ein Wort. — Gespräch: Welches Wort aus eurem Alltag zerfällt wohl in die meisten Stücke?

☐ **Blau 2** — Token 1877 und Token 1878 haben benachbarte Nummern. Was heißt das?
A Nichts, die Nummer ist nur eine Adresse · B Sie bedeuten etwas Ähnliches · C Sie kommen oft hintereinander vor
Die Nummer ist ein Eintrag in einer Liste, keine Aussage. Die Bedeutung steckt ausschließlich in der Position im Bedeutungsraum, nicht in der ID. — Gespräch: Wo verwechselt ihr im Alltag auch schon mal eine Adresse mit dem Inhalt?

☐ **Blau 3** — Warum kannst du ein Modell auf Deutsch, Englisch oder Türkisch prompten?
A Es übersetzt alles zuerst ins Englische · **B Gleiche Bedeutung liegt nah beieinander** · C Für jede Sprache gibt es ein eigenes Modell
Wörter gleicher Bedeutung landen an fast demselben Ort, egal in welcher Sprache. Für seltene Sprachen fehlt das Trainingsmaterial, dort bricht die Qualität ein. — Gespräch: Solltet ihr dann lieber auf Englisch prompten? Was spricht dagegen?

☐ **Blau 4** — Gleicher Inhalt, einmal auf Deutsch, einmal auf Englisch. Welcher Text kostet mehr Tokens?
A Der deutsche, meist etwas mehr · B Der englische · C Beide gleich viel
Komposita und Umlaute zerfallen häufiger in Stücke, bei neueren Modellen weniger als früher. Lange Dokumente füllen so das Kontextfenster schneller und kosten über die API mehr. — Gespräch: Welche eurer Texte sind lang genug, dass das eine Rolle spielt?

☐ **Blau 5** — Wie entsteht die Antwort eines Sprachmodells?
A Es plant den ganzen Satz und schreibt ihn auf · B Es sucht die Antwort in einer Datenbank · **C Es wählt Wort für Wort nach Wahrscheinlichkeit**
Es gibt keinen fertigen Entwurf. Intern schaut es zwar voraus, festgelegt wird aber immer nur ein Wort. Weil jede Wahl den ganzen bisherigen Text berücksichtigt, wirkt das Ergebnis planvoll. — Gespräch: Wie würdet ihr erklären, dass etwas ohne Absicht trotzdem sinnvoll klingt?

☐ **Blau 6** — Du drehst die Temperatur des Modells hoch. Was passiert?
A Es greift öfter zu unwahrscheinlichen Wörtern · B Es wird kreativer und bleibt genauso verlässlich · C Es antwortet schneller
Temperatur ist Zufall, nicht Kreativität. Unwahrscheinlich ist nicht dasselbe wie originell, ab einem gewissen Punkt wird es schlicht wirr. — Gespräch: Für welche eurer Aufgaben wäre mehr Zufall sogar erwünscht?

☐ **Blau 7** — Was passiert mit Text, der aus dem Kontextfenster gefallen ist?
A Das Modell fasst ihn für sich zusammen · B Er wird beim nächsten Mal nachgeladen · **C Er ist für das Modell schlicht nicht vorhanden**
Das Kontextfenster ist eine harte Grenze in Tokens. Was herausfällt, sieht das Modell nicht mehr. Manche Apps fassen vorher zusammen, das tut aber die Anwendung, nicht das Modell. — Gespräch: Wann ist euch ein langer Chat schon einmal spürbar schlechter geworden?

☐ **Blau 8 ★** — Dein KI-Assistent erinnert sich an das Gespräch von letzter Woche. Wer leistet das?
A Das Modell, es lernt aus jedem Gespräch · **B Die Anwendung, über gespeicherte Notizen** · C Niemand, das ist Zufall
Zwischen zwei Anfragen hat das Modell kein Gedächtnis. Die Anwendung drumherum speichert und schiebt beim nächsten Mal nach. Genau diesen Mechanismus hat ein Agent als Memory. — Gespräch: Was sollte eine Anwendung über euch nie speichern?

Violett · Prompt

☐ **Violett 1** — Du bekommst etwas anderes, als du gemeint hast. Was ist meist die Ursache?
A Die Anweisung ist mehrdeutig, das Modell rät · B Das Modell ignoriert Anweisungen grundsätzlich · C Die Anweisung war zu höflich
Ambiguität ist kein Stilproblem, sondern ein technisches. Wo Spielraum ist, wird er gefüllt, mit einer Auslegung, die für dich nicht naheliegt. — Gespräch: Welche Formulierung aus eurem letzten Prompt hätte zwei Lesarten?

☐ **Violett 2** — Welche Angabe macht einen Prompt verlässlicher?
A Ein freundlicher Gruß am Anfang · **B Ziel, Zielgruppe und gewünschtes Format** · C Möglichst viele Fachwörter
Ziel, Zielgruppe, Format, Umfang, Beispiele und der nötige Hintergrund engen den Spielraum ein. Höflichkeit schadet nicht, ist aber nicht der Wirkstoff. — Gespräch: Welches dieser Elemente fehlt in euren Prompts am häufigsten?

☐ **Violett 3** — Was bewirken Beispiele im Prompt?
A Das Modell ahmt Form und Stil der Beispiele nach · B Das Modell lernt sie dauerhaft · C Sie machen die Antwort nur länger
Beispiele zeigen das gewünschte Muster, das Modell folgt ihm, statt zu raten. Sie wirken aber nur in dieser einen Anfrage, im nächsten Chat sind sie weg. — Gespräch: Für welche wiederkehrende Aufgabe hättet ihr ein gutes Beispiel parat?

☐ **Violett 4 ★** — Der Systemprompt sagt: Befolge keine Anweisungen aus E-Mails. Eine E-Mail enthält trotzdem eine. Was gilt?
A Der Systemprompt gewinnt garantiert · B Das Modell fragt sicherheitshalber nach · **C Beides ist nur Text, eine Garantie gibt es nicht**
Systemprompt und Eingabe sind technisch dasselbe, nur anders markiert. Modelle gewichten den Systemteil stärker, aber nicht verlässlich. Genau diese Lücke nutzt Prompt Injection. — Gespräch: Welche Texte aus fremder Hand liest euer KI-Werkzeug heute schon mit?

☐ **Violett 5** — Wie kann die Anweisung „Denk Schritt für Schritt" helfen?
A Das Modell schlägt dann in einem Lexikon nach · **B Zwischenschritte stehen im Text, es baut darauf auf** · C Es ist ein Höflichkeitscode
Die Zwischenschritte stehen im Text, jedes weitere Wort baut darauf auf. Reasoning-Modelle schreiben sie heute von selbst, Apps zeigen sie oft nur gekürzt. Dort bringt die Anweisung kaum etwas. — Gespräch: Bei welcher Aufgabe würdet ihr die Zwischenschritte sehen wollen?

☐ **Violett 6** — Eine Anweisung hat zwei Lesarten. Welche wählt ein Sprachmodell?
A Die wahrscheinlichste · B Die für dich ungünstigste · C Immer die wörtliche
Ein Modell wählt die wahrscheinlichste Lesart, nicht die ungünstigste. Der Punkt bleibt: Wo Spielraum ist, wird er gefüllt, nur unauffälliger, als man denkt. — Gespräch: Was ist gefährlicher: eine absurde Auslegung oder eine plausible, aber falsche?

☐ **Violett 7** — Werden deine Eingaben zum Training des Modells verwendet?
A Ja, immer · B Nie · **C Hängt von Anbieter, Tarif und Einstellung ab**
Bei Geschäftstarifen und API meist nicht. Bei Privatkonten, auch bezahlten, oft schon, solange man nicht widerspricht. Das steht im Vertrag und in den Einstellungen, nicht im Modell. — Gespräch: Wisst ihr, was in eurem Vertrag dazu steht?

☐ **Violett 8** — Gibt es für eine Aufgabe den einen richtigen Prompt?
A Ja, man muss ihn nur finden · **B Nein, nur verlässlichere und unzuverlässigere** · C Ja, der Anbieter liefert ihn mit
Wer eine Formulierung im Betrieb einsetzt, sollte sie an mehreren echten Fällen ausprobieren, statt sie einmal zu bewundern. — Gespräch: Welcher Prompt hat sich bewährt, und an wie vielen Fällen geprüft?

Türkis · Werkzeuge

☐ **Türkis 1** — Was kann ein Sprachmodell ganz von sich aus tun?
A Text erzeugen, sonst nichts · B Text erzeugen und im Internet suchen · C Text erzeugen und Dateien speichern
Das Modell hat keine Hände, auch die Bitte um ein Werkzeug ist nur Text. Wenn ein Chat im Netz sucht, erledigt das der Harness, die Software um das Modell herum. — Gespräch: Welches eurer KI-Werkzeuge tut Dinge jenseits von Text, und wer führt die aus?

☐ **Türkis 2 ★** — Ein Modell ohne Werkzeuge, direkt über die API: Wie spät ist es gerade?
A Es liest die Uhr des Servers · B Es kennt sie aus dem Training · **C Es kann es nicht wissen, höchstens raten**
Das Modell besteht aus festen Gewichten, eine Uhr hat es nicht. Chat-Apps schreiben oft das Datum in den Systemprompt. Über die API muss das der Entwickler tun oder ein Werkzeug anbieten. — Gespräch: Woran würdet ihr in einer Antwort erkennen, dass geraten wurde?

☐ **Türkis 3** — Wer führt ein Werkzeug aus, wenn das Modell darum bittet?
A Das Modell selbst · B Die Nutzerin · **C Der Harness, die Software um das Modell herum**
Das Modell bittet, der Harness entscheidet und führt aus. Er darf auch ablehnen. Genau dort sitzen später Rechte, Limits und Freigaben. — Gespräch: Wo in eurem Unternehmen sitzt heute die Stelle, die ablehnen darf?

☐ **Türkis 4** — Der Werkzeugaufruf des Modells hat einen Tippfehler im Format. Was passiert?
A Der Aufruf greift nicht, nichts wird ausgeführt · B Der Harness korrigiert ihn stillschweigend · C Das Werkzeug läuft trotzdem
Das Format ist technische Voraussetzung, keine Konvention. Gute Harnesses melden den Fehler zurück. Sonst bleibt der Wunsch reiner Text, schlimmstenfalls erfindet das Modell ein Ergebnis. — Gespräch: Was ginge verloren, wenn der Harness großzügig wäre und Formate verhandelbar würden?

☐ **Türkis 5** — Der Harness bekommt ein perfekt formatiertes, aber inhaltlich falsches Ergebnis. Was tut er?
A Er merkt es und stoppt · **B Nichts, er prüft die Form, nicht den Sinn** · C Er fragt das Modell noch einmal
Ein formal korrekter Unsinn läuft durch. Diese Lücke schließt in einem Agenten der Kritiker, und auch der nur teilweise. — Gespräch: Wer prüft bei euch heute Ergebnisse auf Sinn, nicht nur auf Form?

☐ **Türkis 6** — Woher weiß das Modell, welche Werkzeuge es gibt?
A Es kennt sie auswendig aus dem Training · B Es probiert alle der Reihe nach durch · **C Sie werden ihm bei jeder Anfrage mitgeschickt**
Name, Zweck und erwartete Eingaben kommen mit jeder Anfrage mit. Deshalb kosten Werkzeuge Kontext und Geld, und deshalb kann man nicht beliebig viele anbieten. — Gespräch: Welche drei Werkzeuge bräuchte ein Assistent an eurem Arbeitsplatz mindestens?

☐ **Türkis 7** — Kann ein Sprachmodell rechnen?
A Ja, zuverlässig · **B Meist richtig, aber ohne Garantie** · C Nein, nie
Heutige Modelle rechnen oft gut, vor allem mit Zwischenschritten. Garantiert ist es nie, und man sieht es dem Ergebnis nicht an. Viele Apps rechnen deshalb unsichtbar mit einem Werkzeug. — Gespräch: Welche Zahlen aus KI-Antworten habt ihr schon ungeprüft übernommen?

☐ **Türkis 8** — Was ist „Tool Calling“ (früher „Function Calling“) in echter KI-Software?
A Das Modell ruft Programme direkt auf · **B Modell bittet, Software führt es aus** · C Eine Funktion, die das Modell selbst programmiert
Die API liefert Werkzeugaufrufe heute als strukturierte Daten, die Anwendung bindet Werkzeuge oft über den Standard MCP an. Die Formatstrenge ist geblieben, sie ist nur nicht mehr sichtbar. — Gespräch: Seht ihr in euren Werkzeugen, wann ein Werkzeug aufgerufen wurde?

Orange · Agent

☐ **Orange 1** — Was unterscheidet einen KI-Agenten von einem Chatbot?
A Er ist ein größeres Modell · B Er antwortet schneller · **C Er plant, nutzt Werkzeuge, prüft, wiederholt**
Der Agent plant mehrere Schritte, nutzt Werkzeuge, prüft Zwischenergebnisse und läuft in Schleifen, bis das Ziel erreicht ist. Viele Chat-Apps haben dafür inzwischen einen eigenen Modus. — Gespräch: Welche Aufgabe bei euch besteht aus genau solchen Schritten?

☐ **Orange 2** — Ein Agent hat sechs Rollen: Planer, Gedächtnis, Prüfer und drei Werkzeuge. Wie viele Modelle stecken meist dahinter?
A Meist ein einziges, mehrfach anders angewiesen · B Sechs verschiedene · C Keines, das ist klassische Software
Meist ist es dasselbe Modell mit unterschiedlichen Anweisungen, manchmal ein günstiges für Routine und ein starkes fürs Planen. Die Trennung ist Organisation, nicht Hardware. — Gespräch: Welche Rolle würdet ihr dem stärksten, welche dem günstigsten Modell geben?

☐ **Orange 3** — Wozu braucht ein Agent ein Gedächtnis (Memory)?
A Damit das Modell dazulernt · **B Damit Schritt drei weiß, was Schritt eins ergab** · C Damit es schneller antwortet
Memory ist ein Ablageort außerhalb des Kontextfensters, aus dem gezielt etwas zurückgeholt wird. In echten Systemen ist das eine Datei, eine Datenbank oder eine Suche. — Gespräch: Was passiert, wenn Veraltetes nie aus dem Gedächtnis gelöscht wird?

☐ **Orange 4 ★** — Wonach prüft der Kritiker (Critic) im Agenten ein Ergebnis?
A Nach dem Erfolgskriterium der Aufgabe · B Nach Geschmack · C Nach der Länge
Der Kritiker ist selbst ein Modell mit denselben Schwächen. Er hilft, wo ein klares Kriterium steht. Bei der Frage, ob etwas gut ist, hilft er kaum. — Gespräch: Welche eurer Aufgaben hat ein klares Erfolgskriterium, welche nicht?

☐ **Orange 5** — Was passiert ohne Abbruchbedingung in einem Agenten-Loop?
A Er hört nach drei Schritten von selbst auf · B Er liefert dann das beste Ergebnis · **C Er dreht sich im Kreis, teuer und ohne Ergebnis**
Technisch kann ein Agent endlos laufen. Praktisch werden Schrittzahl, Zeit und Kosten begrenzt, sonst merkt niemand sofort, dass er sich im Kreis dreht. — Gespräch: Welche Grenze würdet ihr zuerst setzen: Schritte, Zeit oder Geld?

☐ **Orange 6** — Warum sind die Kosten eines Agenten schwer vorherzusagen?
A Weil die Anbieter die Preise verstecken · **B Weil die Zahl der Schritte von der Aufgabe abhängt** · C Weil jedes Modell anders rechnet
Die Schrittzahl steht vorher nicht fest. Jede Runde kostet einen Modellaufruf, und wie viele es werden, zeigt sich erst beim Laufen. — Gespräch: Wie würdet ihr ein Budget für so etwas festlegen?

☐ **Orange 7** — Zwei Agenten „unterhalten sich“. Was passiert technisch?
A Text geht hin und her, jede Runde kostet · B Sie führen ein Gespräch wie Menschen · C Sie verschmelzen zu einem Modell
Der Nutzen liegt in den unterschiedlichen Blickwinkeln. Der Preis sind Zyklen, jeder davon ein Modellaufruf. — Gespräch: Wann lohnt ein zweiter Blickwinkel die doppelten Kosten?

☐ **Orange 8** — Warum eine Aufgabe in Rollen und Schritte aufteilen, statt alles in einen großen Prompt zu packen?
A Weil große Prompts nicht erlaubt sind · **B Weil man nachvollziehen kann, wo es schiefging** · C Weil es immer billiger ist
Für kleine Aufgaben ist ein einziger Prompt sogar richtig. Ab einer gewissen Länge steigt die Fehlerquote, und vor allem wird prüfbar, an welcher Stelle etwas schiefging. — Gespräch: Welche eurer Aufgaben ist klein genug für einen einzigen Prompt?

Grün · Verantwortung

☐ **Grün 1** — Wie viele Risikoklassen kennt die KI-Verordnung?
A Zwei: erlaubt und verboten · **B Vier: verboten, hoch, Transparenzpflicht, minimal** · C Zehn, je nach Branche
Verboten, Hochrisiko, Transparenzpflicht, minimales Risiko. Die Klasse hängt am Zweck, nicht an der Technik. Transparenzpflichten können zusätzlich greifen, große KI-Modelle haben eigene Regeln. — Gespräch: In welcher Klasse bewegt ihr euch mit euren heutigen Werkzeugen vermutlich?

☐ **Grün 2** — Ein Tool bewertet Bewerbungen und sortiert vor. Welche Risikoklasse?
A Minimales Risiko · B Transparenzpflicht · **C Hochrisiko**
Systeme zur Auswahl von Bewerbungen zählen ausdrücklich zu den Hochrisiko-Anwendungen. Ab Dezember 2027 sind menschliche Aufsicht, Protokollierung und Information der Betroffenen Pflicht. — Gespräch: Wer würde bei euch eine automatische Absage am Ende verantworten?

☐ **Grün 3 ★** — Ihr nutzt nur ein fertiges KI-Werkzeug eines Anbieters. Gilt die KI-Verordnung für euch?
A Ja, als Betreiber habt ihr eigene Pflichten · B Nein, nur für den Hersteller · C Nur ab 250 Mitarbeitenden
Die Verordnung unterscheidet Anbieter und Betreiber. Wer ein KI-System beruflich einsetzt, ist Betreiber. Art. 4 verpflichtet beide, die KI-Kompetenz ihrer Leute aktiv zu fördern. — Gespräch: Wer in eurem Unternehmen müsste das eigentlich auch wissen?

☐ **Grün 4** — Emotionserkennung per Webcam im Meeting, freiwillig per Betriebsvereinbarung. Erlaubt?
A Ja, wenn alle zustimmen · B Ja, mit Kennzeichnung · **C Nein, am Arbeitsplatz verboten, auch freiwillig**
Ein Verbot lässt sich nicht per Einwilligung oder Betriebsvereinbarung aufheben. Ausnahmen gibt es nur aus medizinischen oder Sicherheitsgründen. — Gespräch: Dieselbe Software im Computerspiel wäre harmlos. Was sagt das über die Klassen?

☐ **Grün 5 ★** — Im Prompt steht: „Versende keine E-Mails." Ist das ein Guardrail?
A Nein, das ist eine Bitte, keine Grenze · B Ja, das Modell hält sich daran · C Ja, wenn er in Großbuchstaben steht
Ein Guardrail begrenzt Handlungen technisch oder organisatorisch: Rechte, Betragsgrenzen, Freigaben. Ein Prompt ist nur eine Bitte an ein System, das Wahrscheinlichkeiten berechnet. — Gespräch: Welche echte Grenze hat euer KI-Werkzeug heute?

☐ **Grün 6** — Wann muss ein Mensch in den Loop?
A Immer, bei jedem einzelnen Schritt · **B Bei hohem Einsatz oder schwer umkehrbaren Folgen** · C Nie, sonst lohnt sich der Agent nicht
Kriterien sind Schadenshöhe, Umkehrbarkeit und ob der Agent eigene Fehler erkennt. Die Stufe hängt an der einzelnen Handlung, nicht am Produkt: unter 50 Euro allein, darüber Freigabe. — Gespräch: Welche Handlung in eurem Alltag wäre die erste, die ihr freigeben lassen würdet?

☐ **Grün 7** — Ein Bestätigungsklick vor jeder Aktion des Agenten. Ist das Kontrolle?
A Ja, immer · **B Nur mit Information, Zeit zum Prüfen und echtem Nein** · C Nein, nie
Ein Dialog, den man fünfzig Mal am Tag wegklickt, ist Dokumentation, nicht Kontrolle. Für Hochrisiko-Systeme verlangt die Verordnung wirksame Aufsicht und warnt ausdrücklich vor Automatisierungsbias. — Gespräch: Welche Bestätigung klickt ihr heute schon weg, ohne zu lesen?

☐ **Grün 8** — Ein voll autonomer Agent trifft eine Fehlentscheidung. Wer trägt die Verantwortung?
A Der Agent · B Der Anbieter des Modells · **C Das Unternehmen, das ihn einsetzt**
Autonomie verschiebt die Arbeit, nicht die Verantwortung. Am Ende steht ein Mensch oder eine Organisation gerade, in aller Regel nicht die Person, die den Knopf gedrückt hat. — Gespräch: Wer unterschreibt bei euch, wenn der Agent entscheidet?

Lizenz und Namensnennung

Fragen-Bingo – Vom Token zum Agenten von Peter Schmidt (versabilit.com), Stand Oktober 2026. Lizenz: CC BY-SA 4.0. Du darfst das Material teilen und bearbeiten, auch kommerziell. Bedingung: Namensnennung mit Quelle und Lizenz, Änderungen kennzeichnen, Bearbeitungen unter derselben Lizenz weitergeben.

So nennst du die Quelle: *„Fragen-Bingo – Vom Token zum Agenten“ von Peter Schmidt (versabilit.com), CC BY-SA 4.0.* Bei einer Bearbeitung: *Basiert auf „Fragen-Bingo – Vom Token zum Agenten“ von Peter Schmidt (versabilit.com), CC BY-SA 4.0; verändert von …*

KI-Hinweis. Erstellt mit KI-Unterstützung (Claude, Anthropic) für Icons, Recherche und Formulierungen, geprüft und verantwortet vom Urheber.

Haftung. Keine Rechtsberatung. Rechtsangaben Stand Oktober 2026, nicht fachlich geprüft.

Schriften: Inter und JetBrains Mono, beide unter der SIL Open Font License 1.1.