Vom Wunsch zum prüfbaren Fachagenten

Ein Blick in den Fachagenten-Builder Skill: seine Originalregeln, ein Beispiel aus dem Testlauf und das, was die Tests belegen und was nicht.

Maximilian Tabor

Fachagenten-Builder · Skill- & Testdokumentation · Version 1.1.0

KURZ-ZUSAMMEN-FASSUNG

Der Fachagenten-Builder ist kein Generator, der auf Zuruf einen Prompt ausgibt. Er ist ein Skill mit drei Arbeitsmodi (Bauen, Architektur prüfen, Testen & Optimieren), einem Router auf sieben Referenzdokumente und harten Schranken, etwa einer Systemanweisung von höchstens 7.999 Zeichen und dem Verbot, 9 von 10 Punkten ohne Tests zu behaupten. Das Beispiel eines Reklamations-Assistenten zeigt, wie der Skill Quellen stuft, Lücken markiert und Freigaben einbaut. Die Tests zeigen einen deutlichen Vorteil gegenüber demselben Modell ohne Skill. Sie fanden aber in einer Ersatzumgebung und mit kleiner Fallzahl statt und sind kein Nachweis für den Einsatz auf der Zielplattform.

3	7	26/26	0/8
Arbeitsmodi	Referenzen A bis G	Prüfpunkte, v1.0, vier Läufe	Läufe, in denen das Playbook gelesen wurde

01 Die zentrale Frage ist nicht, was ein Agent tut, sondern woran man erkennt, dass er es gut tut

Ein Fachagent für Büroarbeit lässt sich schnell behaupten. Schwerer ist es zu klären, welche Aufgabe er genau übernimmt, wie gut er sein muss, woher sein Wissen stammt, wer freigibt und wie man das alles prüft. Der Fachagenten-Builder macht genau diese Fragen zum Ablauf.

Der Skill unterscheidet dafür drei Situationen. Wer einen neuen Agenten will, wird gebaut. Wer einen vorhandenen Aufbau bewerten lässt, bekommt eine Architekturprüfung. Wer Verhalten, Testfälle oder Verbesserungen verlangt, landet im Testmodus. Dabei lädt er je Auftrag nur die Referenzen, die er braucht.

Die Regeln sind auf mehrere Dateien verteilt. Der Kern bleibt kurz, das Tiefenwissen liegt in sieben Referenzen. Das ist eine bewusste Architekturentscheidung: Die Abläufe lassen sich einzeln prüfen und ändern, ohne den Kern anzufassen.

02 Die Skillstruktur in vier Ebenen

EBENE 1 · MODUS UND VERANTWORTUNG

Zuerst entscheidet der Skill, welche Arbeit gemeint ist. Das verhindert, dass ein Bauwunsch mit einer Testfrage vermischt wird. Ein Originalausschnitt:

KARTE 01 · MODUS BESTIMMEN — SKILL.md

```
- Neuer Agent / Skillpaket / ZIP -> **Bauen**.
- Vorhandenen Aufbau bewerten -> **Architektur pruefen**.
- Verhalten, Testfaelle, Bewertung oder Verbesserungen -> **Testen & Optimieren**.
- Mehrere Anliegen -> sinnvoll sequenzieren und die Abhaengigkeiten nennen.
```

Auch Mischaufträge bekommen eine Reihenfolge. Die Abhängigkeiten müssen benannt werden.

EBENE 2 · REFERENZ-ROUTER

Der Skill lädt nicht alles, sondern wählt nach Aufgabe. Jede Referenz hat einen eigenen Zweck, vom Auftrag (A) bis zur Paketierung (G):

KARTE 02 · REFERENZ-ROUTER — SKILL.md

```
- **A** fuer Ziel, Nutzer, Kontext, Beispiele und Qualitaetsmassstab.
- **B** fuer Aufgabencluster, Router, Handlungspfade und Modulgrenzen.
- **C** fuer Systemprompt, Skill, Playbooks, Referenzarchitektur und Zeichengrenze.
- **D** fuer Web-/Integrationsrecherche, Quellenregeln und Mensch-Builder-Aufgabenteilung.
- **E** fuer Testmatrix, Szenarien, Testmodus und Testprotokoll.
- **F** fuer Bewertung, Ursachenanalyse, priorisierte Optimierung und Regression.
- **G** fuer Paketstruktur, Manifest, technische Checks und doppelte Abschlusspruefung.
```

Der Router verhindert, dass ein Test mit Bauwissen beantwortet wird oder umgekehrt. Er hat eine Schwäche, die Abschnitt 04 zeigt: Verweist kein Arbeitsschritt auf eine Datei, wird sie nicht gelesen.

EBENE 3 · REGELN FÜR DEN KONKRETEN ARBEITSSCHRITT

Jeder der drei Modi hat einen eigenen Ablauf: sieben Schritte beim Bauen, acht beim Testen und Optimieren. Dazu kommen Regeln in den Referenzen. Zwei Beispiele zeigen den Ton: Fragen werden gedeckelt, und Wissen darf nur an einer Stelle gelten.

KARTE 03 · FRAGEPRINZIP — A_AUFTRAG_UND_EXZELLENZ.md

```
Nutze vorhandene Chat- und Dateiinformationen zuerst. Stelle nur Fragen, deren Antwort den Agentenaufbau
veraendert. Pro Runde hoechstens drei entscheidende Fragen.
```

Im Test „Baue mir einen Sales-Agenten“ hat der Skill genau 3 Fragen gestellt und nichts gebaut. Ohne Skill waren es 7 Fragen.

KARTE 04 · EINE AUTORITÄT PRO REGEL — B_ARCHITEKTUR_UND_ROUTING.md

```
Lege Inhalt nur einmal als Autoritaet fest. Wenn zwei Module Regeln enthalten, benenne Vorrang und
Konfliktaufloesung.
```

Doppelte Regeln gelten als Fehlerquelle. Im Testfall „defekter Agent“ war ein doppelter Ablauf einer der gefundenen Mängel.

EBENE 4 · SCHRANKEN, UNSICHERHEIT UND FREIGABE

Die Schranken stehen im Kern und nicht nur in optionalen Dateien. Sie bestimmen, was der Skill gar nicht erst tun darf:

KARTE 05 · GLOBALE SCHRANKEN — SKILL.md

```
- Systemanweisung <8.000 Zeichen; maximal 7.999 inkl. Leerzeichen/Zeilenumbruechen.
- Keine 9/10-Behauptung ohne Tests und nachvollziehbare Belege.
- Keine erfundenen Quellen, Laufzeitergebnisse, Referenzinhalte oder Unternehmensregeln.
- Bei nicht verfuegbarer Recherche oder Ausfuehrungsumgebung: Grenzen benennen und sicheren Ersatz
liefern.
```

Das ist keine Warnung am Ende der Antwort. Die Schranken legen fest, was der Skill von Anfang an ausführen darf.

KARTE 06 · TESTEBENEN — E_TESTDESIGN_UND_TESTMODUS.md

```
1. Statische Analyse: ... Kein Verhaltenstest.
2. Schreibtischsimulation: ... Nicht als echter Laufzeittest ausgeben.
3. Laufzeittest: tatsaechliche Interaktion mit bereitgestelltem Agenten/System.
```

Jedes Testergebnis muss seine Ebene nennen. So wird eine Simulation nicht mit einem echten Lauf verwechselt.

03 Ein konkretes Beispiel aus dem Testfall

Der Testfall: Der Skill soll einen Reklamations-Assistenten für einen Haushaltsgeräte-Shop bauen. Die Freigaben waren vorab erteilt, weil im Test niemand Rückfragen beantworten konnte. Der Skill arbeitete den Ablauf ab, recherchierte und prüfte sein Paket selbst.

EINGABE · BAUAUFTRAG — TESTLAUF reklamation-vollbau-recherche

```
Reklamations-Agent für einen Haushaltsgeräte-Shop, Vorab-Freigabe erteilt.
```

PRÜFPUNKT	OHNE SKILL	MIT SKILL
Quellenverzeichnis	fehlt	20 Quellen (Stufe A 14, B 4, C 2)
Evidenzstufen	fehlen	vergeben, A (gilt als Regel) bis D (Hypothese)
Norm ISO 10002	nicht berücksichtigt	einbezogen, abgestuft, nur als Abstract zugänglich
Zeichengrenze der Systemanweisung	nicht gezählt	gezählt: 5.124 von 7.999 Zeichen
Nicht verifizierbarer BGH-Hinweis	im Protokoll nicht erfasst	**weggelassen**
Ergebnis insgesamt	**3 von 7**	**7 von 7**

Der Skill recherchierte mit 10 Suchen und 14 geöffneten Seiten. Das Paket umfasste 16 Dateien und bestand das Prüfskript. Wichtig ist, was bewusst nicht behauptet wurde: Die Norm war nur im Abstract lesbar und wurde so gekennzeichnet. Der BGH-Hinweis war nicht verifizierbar und blieb draußen.

Im späteren Testlauf wurde der gebaute Assistent selbst geprüft, mit einem erfundenen Fall: Eine Waschmaschine, geliefert am 02.06.2026, Reklamation am 10.11.2026, also nach rund fünf Monaten. Der Entwurf bezog sich auf § 477 BGB (Beweislastumkehr) und § 439 Abs. 1 BGB (Wahlrecht der Nacherfüllung) und stufte den Fall als „berechtigt, Sicherheit mittel“ ein. Interne Entscheidungen blieben als Platzhalter offen:

PRÜFBLATT · PLATZHALTER IM ENTWURF antwort.md

```
[ENTSCHEIDUNG TEAM: ...]
[FEHLT: Rolle ...]
```

Der Assistent erfindet keine Unternehmensregel. Er markiert, was das Team entscheiden muss.

04 Was die Tests zeigen und was nicht

Getestet wurde in zwei Runden. Jeder Test lief in einem eigenen, frischen Hilfsagenten, einmal mit und einmal ohne Skill. Version 1.0 lief in vier Läufen, Version 1.1 wurde danach gegen die Vorversion getestet.

LAUF	MIT SKILL	AUFFÄLLIGSTER UNTERSCHIED
Sales-Agent mit fehlenden Angaben	5 von 5	3 Fragen und kein Bau, ohne Skill 7 Fragen
Reklamations-Agent mit Recherche	7 von 7	Quellenverzeichnis, Evidenzstufen, gezählte Zeichen
Defekter Testagent	7 von 7	2,6 von 10, drei kritische Fehler, eingeschleuste Anweisung gemeldet und nicht befolgt
Interne Daten, SharePoint	7 von 7	Keine erfundene Quote, stattdessen `[FEHLT: …]`
v1.0 gesamt	26 von 26	ohne Skill 49 % der Prüfpunkte
v1.1 gegen v1.0	20 von 20	v1.0 erfüllte 11 von 15 Prüfpunkten
Weichen-Test, 9 knappe Fehlgriffe	20 von 20	Alle 20 Anfragen richtig geroutet

WAS DIE LÄUFE ÜBER DEN SKILL ENTHÜLLT HABEN

Der wichtigste Fund war kein Erfolg, sondern ein Befund über die Architektur, nicht über den Nutzen von Recherche: Das Fach-Playbook des gebauten Assistenten wurde in 0 von 8 Läufen geöffnet. Es gab drei Ursachen. Kein Arbeitsschritt verlangte die Datei. Das Wissen stand bereits in zwei anderen Dateien. Und eine Regel verwies auf eine leere Datei, weshalb die Kostenaussage im Kundenentwurf von Lauf zu Lauf schwankte. Außerdem waren die Prüfpunkte zu leicht: 100 % gegen 100 % trennen nichts.

Version 1.1 reagiert darauf. Fachwissen wird an den Entscheidungspunkten der Pfade verankert, leere Dateien bekommen ein festes Verhalten, und jeder Lauf protokolliert, welche Dateien er gelesen hat. Zwei Fehler im eigenen Prüfskript wurden dabei gefunden und korrigiert.

Was heute belegt ist und was noch fehlt.

- **Ersatzumgebung:** Alle Läufe fanden dort statt, und laut Builder ist sie eher zu optimistisch. Genau deshalb verbietet der Skill 9 von 10 Punkten ohne Tests auf der Zielplattform. Die Ergebnisse sind Entwicklungsbefund, kein Produktivnachweis.
- **Fallzahl:** Sie ist klein, Streuung über viele Wiederholungen wurde nicht gemessen. Aussagen gelten nur für die getesteten Fälle.
- **Version 1.1:** Die Änderungen nach dem Playbook-Befund sind eingebaut, aber noch nicht erneut gelaufen. Der Nachweis steht aus.
- **Ungetestet:** 8.200-Zeichen-Entwurf, schwache Evidenz, Betrieb ohne Websuche und die Frage, ob der Skill von selbst ausgelöst wird.
- **Vergleich gegen freien Chat:** Er wurde vom Autor noch nicht durchgeführt. Ob das Modell den Skill im Alltag nutzt, ist offen.
- **Aufwand gegen Ertrag:** Ein Vollbau mit Recherche braucht rund 170.000 Tokens, ohne Skill rund 70.000. Der Mehraufwand bezahlt Quellen, Evidenzstufen, gezählte Zeichen und Prüfprotokolle, ohne Skill fehlten sie. Ein Lauf mit 20 Durchgängen kostet rund 200.000 Tokens und 10 bis 13 Minuten.

Die Ergebnisse sind deshalb ein nützlicher Entwicklungsbefund, aber kein allgemeiner Leistungsnachweis.

05 Was sich aus dem Skilldesign lernen lässt

NR.	KERNPRINZIP FÜR PRÜFBARE FACHAGENTEN
01	**Wer baut, prüft nicht.** Der Skill trennt Bauen und Prüfen. Danach testet ein zweiter Agent in getrenntem Kontext, und die Logik wird separat bewertet.
02	**Wissen gehört an genau eine Stelle.** Wo zwei Module Regeln enthalten, muss der Vorrang benannt sein. Doppelte Quellen führen zu Widersprüchen.
03	**Fehlendes ist ein Ergebnis.** Fehlende interne Zahlen werden als `[FEHLT: …]` markiert und nicht geschätzt. Ein gutes System muss nicht jede Lücke schließen, es muss sie benennen.
04	**Quellen haben Rang.** Gesetze und Normen zählen mehr als Fachverbände, Meta-Analysen mehr als Einzelstudien, Standardwerke mehr als Blogs. Die Stufen reichen von A (gilt als Regel) bis D (Hypothese).
05	**Eine Datei, die kein Arbeitsschritt verlangt, wird nicht gelesen.** Wissen muss dort stehen oder verlinkt sein, wo die Entscheidung fällt.
06	**Tests müssen die richtige Frage prüfen.** Ein Test, den beide Varianten bestehen, trennt nichts. Auch die eigene Testumgebung gehört kritisch gelesen.